

SEP

MyDCI

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA.

FACULTAD DE INGENIERÍA.



**HERRAMIENTAS PARA DETERMINACIÓN DE REGULARIDAD
DE MOVIMIENTO ARTICULADO MEDIANTE EL
SEGUIMIENTO EN UNA SECUENCIA DE IMÁGENES.**

**Trabajo de tesis que para cubrir parcialmente los requisitos para obtener
el grado de MAESTRO EN CIENCIAS presenta:**

Ing. Jesús Ernesto Valdez Almaral.

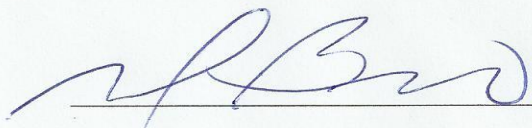
Director de Tesis:

Dr. Miguel Enrique Bravo Zanoguera.

RESUMEN de la tesis de Ing. Jesús Ernesto Valdez Almaral, presentada como requisito parcial para la obtención del grado de MAESTRO EN CIENCIAS. Mexicali, Baja California, México. Abril de 2014.

Herramientas para la detección de regularidad de movimiento articulado mediante el seguimiento en una secuencia de imágenes.

Resumen aprobado por:



Dr. Miguel Enrique Bravo Zanoguera
Director de tesis

El presente trabajo de tesis desarrolla métodos de obtención, análisis y comparación de las trayectorias que siguen objetos que aparecen en una secuencia de video para evaluar su regularidad. Para ello se han creado un conjunto de algoritmos, a modo de librería, que genera primero la trayectoria seguida por un objeto o persona, para después extraer parámetros de esta trayectoria, y luego comparar estos parámetros y determinar qué tan similar es una trayectoria con respecto a otra de referencia; los parámetros sirven también de atributos en algoritmos de clasificación. La caracterización de trayectorias y la evaluación de su regularidad pueden ser utilizadas en áreas tan diversas como la rehabilitación física (diagnóstico médico al evaluar trayectorias de movimientos de pacientes sanos y lesionados), la climatología (Predicción de trayectorias de huracanes mediante generación de bases de datos de trayectorias seguidas por fenómenos anteriores). También son pertinentes en tareas como monitoreo industrial y/o vial (Evaluación de trayectorias seguidas por un mecanismo o vehículo para saber si existe alguna irregularidad que salga de sus movimientos considerados como normales) y los deportes (Análisis de trayectorias de deportistas en tareas repetitivas como saltos y clavados).

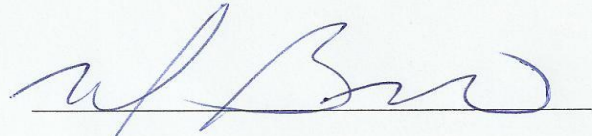
Para la experimentación con los algoritmos diseñados se utilizaron tres conjuntos de videos, el primero consistió en un robot de cinco grados de libertad ejecutando una rutina de movimiento; el segundo en una serie de videos de un motor con movimiento de vaivén tipo péndulo y el tercer conjunto consistió en una serie de videos de una gimnasta al hacer dos tipos diferentes de saltos. Como resultados de la aplicación de las herramientas de obtención, caracterización y comparación de trayectorias, se obtuvo una serie de trayectorias de los tres conjuntos de videos, y en cuyo procesamiento se obtuvo un error en la detección del objeto de ± 1.3048 pixeles en X y de ± 1.95 pixeles en Y. Lo anterior sin la necesidad de utilizar ningún tipo de equipamiento especial. Además se hizo una

comparación de trayectorias de dos tipos de salto diferentes ejecutados por una gimnasta, obteniendo un parámetro de evaluación ε para cada comparación, cuyo valor resulta tener rango suficiente para establecer un umbral que identifica las trayectorias que son semejantes. Además de parámetros tradicionales, esta investigación plantea un nuevo parámetro de comparación de similitud de trayectorias, denominado *ángulo de dirección de objeto*, el cual mostró resultados congruentes con la similitud entre las trayectorias comparadas, entre menor sea la diferencia del ángulo de dirección de dos trayectorias, más similares son entre sí. Para el caso de los saltos distintos se tuvo una diferencia en el ángulo de dirección de 18.2 grados, mientras que para el caso de la comparación con trayectorias similares, se tuvo una diferencia en el ángulo de dirección de 0.56 grados.

ABSTRACT of the thesis by Jesús Ernesto Valdez Almaral, presented in order to obtain the degree of MASTER IN SCIENCE. Mexicali, Baja California, México. April 2014.

Tools for the detection of articulated movement regularity through object tracking in an image sequence.

Abstract approved by:



Dr. Miguel Enrique Bravo Zanoguera
Thesis Director

ABSTRACT

This thesis develops methods to obtain, analyze and compare a path that follow an object that appear in a video sequence, in order to evaluate its regularity. For this purpose, a library of algorithms have been created, to firstly create the path followed by an object or person, then extract parameters of this path, and then compare these parameters and determine the similarity of one path with respect to a reference path; the parameters also serve as attributes in classification algorithms. The characterization of paths and the evaluation of its regularity can be used in areas as diverse as physical rehabilitation (medical diagnosis when evaluating movement of healthy and injured patients), weather (forecasting of trajectories of hurricanes through assembly of trajectories from previous phenomena databases). They are also relevant in tasks as industrial or road monitoring (evaluation of trajectories followed by a mechanism or vehicle to find any irregularity in their movements) and sports (analyzing trajectories of athletes in repetitive tasks such as Olympic jumps and dives).

Three sets of videos were used for experimentation with the algorithms, the first video consisted of a robot with five degrees of freedom working a routine program; the second video was a pendulum type engine with a back and forth motion, and the third set consisted of videos of a gymnast doing two different types of Olympic jumps. The applied tools for collection, characterization and comparison of paths returned all the paths of the three sets of videos, but with an error in the detection of the object of ± 1.3048 pixels in X

and ± 1.95 pixels in Y. This was obtained without the need for any special hardware. In addition, a comparison of trajectories of two different types of jumps executed by a gymnast was done, obtaining an evaluation parameter ε for each comparison, and whose value had sufficient range to establish a threshold that identifies paths that are similar. In addition to traditional parameters of comparison, in this research we introduced a new parameter to find similarity of trajectories, called *object angle*, which showed consistent results with the similarity between the trajectories evaluated, as the difference of this angle parameter is lower, more alike are the trajectories. For the case of different jump trajectories the difference in the *angle of direction* was 18.2 degrees, while in the case of the comparison of similar trajectories, the difference in the *angle of direction* was of 0.56 degrees.

Índice

	<u>Página</u>
CAPÍTULO 1 INTRODUCCIÓN	1
1.1 Planteamiento del problema	1
1.2 Objetivos	4
1.2.1 Objetivos Generales	4
1.2.2 Objetivos particulares	4
1.3 Justificación	5
1.4 Hipótesis	6
1.5 Alcances	7
1.6 Limitaciones	7
1.7 Metodología	8
1.8 Estructura de la tesis	9
CAPÍTULO 2 ESTADO DEL ARTE, MARCO TEÓRICO Y ANTECEDENTES	11
2.1 Estado del arte	12
2.1.1 Estado del arte en tecnología	12
2.1.2 Estado del arte en medición de regularidad de movimiento y comparación de trayectorias	14
2.2 Representación de imágenes digitales	15
2.3 Selección de características para seguimiento de objetos	18
2.4 Detección de objetos	20
2.5 Ajuste de curvas	23
2.5.1 Redes Neuronales	23
2.5.2 Ajuste de curvas mediante mínimos cuadrados	24
2.5.3 El error cuadrático medio	24

2.6	Trabajo base	25
2.7	Bases de datos	27
2.8	Contexto de la investigación	29
2.9	Diagrama general del proceso total	29

CAPÍTULO 3 IMPLEMENTACIÓN DE ALGORITMOS PARA OBTENCIÓN DE TRAYECTORIAS Y AJUSTE DE CURVAS

31

3.1	Separación de un video en imágenes para su procesamiento	31
3.2	Determinación de la intensidad de colores rojo, verde y azul (RGB) de una imagen en un área de interés	32
3.2.1.	Obtención del histograma de las regiones de interés	32
3.3	Descripción del proceso de obtención de trayectorias	36
3.4	Problemas encontrados durante el seguimiento y la generación de trayectorias	43
3.5	Comparación de técnicas usadas para el ajuste de trayectorias	48
3.5.1.	Implementación de una red neuronal para ajuste de curvas de trayectorias	48
3.5.2.	Implementación de algoritmo de mínimos cuadrados para ajuste de curvas de trayectorias	54
3.5.3.	Comparación de resultados de algoritmos de ajuste de curvas de trayectorias	56

CAPÍTULO 4 EFECTO DE LA ILUMINACIÓN EN LA DEFINICIÓN DE CURVAS DE TRAYECTORIA Y EXTRACCIÓN DE PARÁMETROS DE SIMILITUD PARA COMPARACIÓN ENTRE TRAYECTORIAS

59

4.1	Determinación de la importancia de una iluminación constante en la captura de videos	59
4.1.1.	Medición de la intensidad de iluminación del entorno de trabajo	60
4.1.2.	Resultados de las mediciones y trayectorias a diferentes grados de iluminación	61
4.2	Análisis del error de captura de centroides en videos de motor de vaivén	65
4.3	Extracción de parámetros de similitud de curvas de trayectorias	67

4.3.1.	Descripción de parámetros de similitud de trayectorias	67
4.3.2.	Obtención del parámetro de ángulo de dirección de objeto	70
4.3.3.	Comparación de parámetros de similitud para videos de saltos de gimnasta	76
4.3.4.	Parámetro de evaluación para determinación de similitud entre trayectorias	78
CAPÍTULO 5 DISCUSIÓN, CONCLUSIONES Y TRABAJOS FUTUROS		81
5.1	Discusión y conclusiones	81
5.1.1	Conclusión sobre la técnica seleccionada para ajustes de curva de trayectoria	81
5.1.2	Conclusión sobre la obtención del parámetro de evaluación de similitud ϵ	82
5.1.3	Ventajas en el uso de las herramientas diseñadas en comparación con otros sistemas y/o métodos	82
5.1.4	Innovaciones y mejoras	84
5.1.5	Conclusiones sobre las pruebas de iluminación	86
5.1.6	Restricciones, limitantes y contexto de uso	87
5.1.7	Posibles aplicaciones	90
5.2	Trabajos futuros	91
BIBLIOGRAFÍA		92
APÉNDICE A GLOSARIO DE TÉRMINOS Y CONCEPTOS		98
A.1.	Conceptos generales	98
A.2.	Histograma y sus propiedades estadísticas	101
A.3.	Mecanismos de detección de objetos	104
A.3.1	Detectores puntuales	104
A.3.2	Substracción de fondo	105
A.3.3	Segmentación	105
A.4.	Definición de red neuronal	107
APÉNDICE B ALGORITMOS DISEÑADOS		109
B.1.	Algoritmo para generación de trayectorias (edp_AI).	109

B.2.	Función de asignación de umbral para videos de salto	111
B.3.	Función de selección de puntos para eliminar falsos positivos	112
B.4.	Algoritmo de entrenamiento de red neuronal para ajuste de curvas	113
B.5.	Función de interpolación mediante mínimos cuadrados	115
B.6.	Función de evaluación de función de interpolación	117
B.7.	Función de obtención de error cuadrático medio	118
B.8.	Función de correlación y covarianza cruzadas	119
B.9.	Función de obtención de puntos críticos	120
B.10.	Función de obtención de puntos de inflexión	122
B.11.	Función de obtención de velocidades	124
B.12.	Función de obtención de índice de rectitud (ST)	126
B.13.	Función de obtención de ángulos de dirección de objetos	127
B.14.	Función de generación de fondo promediado	132
B.15.	Función de umbralado automático Ridler Calvard	133
B.16.	Función de promediado de datos en coordenadas en Y	134
B.17.	Algoritmo principal	136
APÉNDICE C TABLAS DE DATOS (X,Y,T) PARA VIDEOS DE SALTOS		141
C.1.	Tablas de datos de videos de saltos para un sub muestreo de aproximadamente 100 imágenes	141
C.1.1.	Tabla de datos salto A	141
C.1.2.	Tabla de datos salto B	142
C.1.3.	Tabla de datos salto C	143

C.2. Tabla de coordenadas de centroides correspondientes al *target* central en el mecanismo de motor de vaivén para el video de todas las lámparas encendidas

144

Índice de Figuras

Figura 2.1. Representación matricial de una imagen digital en escala de grises. La matriz del lado derecho está dada por el valor de intensidad de los píxeles en el recuadro blanco que está entre la parte superior del árbol y el cielo de la imagen del lado izquierdo.	16
Figura 2.2. Representación matricial de una imagen a color RGB, los cuadros negros del lado izquierdo de la imagen son regiones de 3x3 píxeles, y cada región está dada por tres matrices $R(X,Y)$, $G(X,Y)$, $B(X,Y)$.	17
Figura 2.3. Representación matricial de una imagen binaria, cada cuadro de la imagen del lado izquierdo es tomado como un pixel.	17
Figura 2.4. Segmentación de una imagen por medio de detección de umbral. a) Imagen original, b) Imagen resultante después de haber aplicado el método de segmentación con asignación de umbral (Imágenes extraídas de Q. Sánchez, "Estimación de Movimiento de Objetos a Partir de una Secuencia de Imágenes", [33]).	21
Figura 2.5. Resultado de la figura 2.4 después de haber aplicado proceso de etiquetado.	22
Figura 2.6. Aplicación de una red neuronal para un problema de regresión en polinomios (Imagen extraída de C. M. Bishop, "Neural Networks for Pattern Recognition", [34]).	23
Figura 2.7. Objetos seguidos en el trabajo de tesis de Juan Carlos Quiroz [33], (a) conjunto de hormigas, (b) robot GE-Fanuc LR MATE 200i.	25
Figura 2.8. Resultados gráficos de la trayectoria seguida por un objeto, (a) Grupo de hormigas, (b) Brazo robótico GE-Fanuc LR MATE 200i [33].	26
Figura 2.9. Videos utilizados como bases de datos para experimentación. (a) Motor con movimiento de vaivén tipo péndulo. (b) Gimnasta realizando un salto (ver recuadro amarillo).	27
Figura 2.10. Diagrama general del proceso total de la presente tesis.	30
Figura 3.1. Histograma de una región de una imagen de los videos de péndulo en Adobe Photoshop. Se observan los tres histogramas de color y sus medidas estadísticas	33
Figura 3.2. Histograma de una región de una imagen de los videos de salto en Adobe Photoshop®.	34

Figura 3.3. a) Motor de CD con movimiento de vaivén, b) Imagen binaria obtenida de la segmentación, c) Imagen binaria con regiones de interés rellenadas, d) Centroides graficados de las regiones de interés.	38
Figura 3.4. Grafica de todos los puntos centroides que deben corresponder a la trayectoria seguida por un grupo de objetos asignados a un motor con movimiento de vaivén.	40
Figura 3.5. Gráfica de la trayectoria seguida por una gimnasta al realizar un salto. Trayectoria con sub-muestreo de 100 imágenes.	41
Figura 3.6. Gráfica aislada de la trayectoria seguida por una gimnasta al realizar un salto. Trayectoria sin sub-muestreo.	41
Figura 3.7. Gráfica de las trayectorias seguidas por la gimnasta en tres saltos.	42
Figura 3.8. Ejemplos de imágenes con detección fallida (falsos negativos).	45
Figura 3.9. Muestra de trayectoria con falsos positivos.	46
Figura 3.10. Ilustración del algoritmo interactivo selectPoints_Al para eliminación de falsos positivos. a) Trayectoria antes de aplicar algoritmo, b) trayectoria después de aplicar algoritmo.	47
Figura 3.11. Trayectoria del motor de vaivén graficada en coordenadas (X, Y). Trayectoria obtenida utilizando un sub-muestreo de 50 imágenes.	49
Figura 3.12. Ajuste de trayectoria de motor con movimiento de vaivén y su comparación con los puntos de referencia.	52
Figura 3.13. Ajuste de trayectoria seguida por gimnasta al realizar un salto mediante red	53
Figura 3.14. Ajuste mediante mínimos cuadrados de trayectoria seguida por gimnasta al realizar un salto.	55
Figura 4.1. a) Mesa “Polaroid MP4 Land Camera”, cuenta con cuatro lámparas y altura ajustable para ubicar una cámara. b) Esquema simple de mesa polaroid MP4 vista desde arriba. L1 a L4 representa la posición física de cada lámpara que contiene la mesa.	59
Figura 4.2. a) Luxómetro usado en pruebas de iluminación. b) Celda fotoeléctrica colocada en el centro de la mesa Polaroid para medición de iluminación.	60
Figura 4.3. Trayectoria del motor tipo péndulo y su comparación consigo misma para diferentes configuraciones de iluminación. a) Todas las lámparas encendidas, b) Todas las lámparas apagadas, c) Lámpara L2 apagada, d) Lámpara L1 apagada, e) Lámparas L1 y L2 apagadas, f) Lámparas L2 y L4 apagadas	64

Figura 4.4. Acercamiento a los centroides correspondientes al <i>target</i> central del mecanismo de motor de vaivén en el video de todas las lámparas encendidas, el círculo negro es la media de los datos.	66
Figura 4.5. Planos anatómicos humanos (Imagen extraída de Moore L., “Anatomía con orientación clínica”, [36]).	70
Figura 4.6. a) Imagen pre procesada. b) Imagen con fondo recortado. c) Imagen con substracción de fondo. d) Detección de gimnasta mediante algoritmo de Ridler Calvard. e) Imagen de gimnasta después de ser procesada.	71
Figura 4.7. Imagen de ejemplo con componentes principales encontrados en la gimnasta.	73
Figura 4.8. Ilustración del ángulo de dirección de objeto.	73
Figura 4.9. Ejemplo de obtención de ángulos de dirección de objeto para una serie de cuadros de video de gimnasta.	74
Figura 4.10. Trayectorias de salto de gimnasta con parámetros de similitud de mínimos (círculos negros), máximos (asteriscos azules) y puntos de inflexión (cuadros rojos).	76
Figura 5.1. Apreciación objetos para distintas posiciones. a) <i>Target</i> bien definido, b) Objeto segmentado con forma cercana a un círculo, c) <i>Target</i> difuminado, d) Objeto segmentado con forma irregular	88
Figura 5.2. Centroides percibidos en posiciones distintas debido a la deformación del target utilizado para detección, a) Posición 1, b) Posición 2.	89
Figura A.1. Trayectoria de un objeto en movimiento: Representación espacio-temporal del movimiento de un objeto (Imagen obtenida de S. Dodge, “Exploring Movement Using Similarity Analysis”, [16]).	100
Figura A.2. Histograma representativo de una imagen en escala de grises (izquierda) y una imagen a color RGB (derecha).	102
Figura A.3 Ejemplo de detección de objetos mediante detectores puntuales (Imagen tomada de P. Gabriel, “Object Tracking Using Color Interest Points”, [42]).	104
Figura A.4. Ejemplo de método de substracción de fondo para detección de objetos (Imagen tomada de L. Ssu-Wei Chen, “Moving Object tracking Based on Background Subtraction Combined Temporal Difference”, [43]).	105

Figura A.5. Ejemplo del método de segmentación para detección de objetos (Imagen tomada de “W. K. Pratt, Digital image processing”, [41]).	106
Figura A.6. Representación de una neurona en forma artificial [49].	107
Figura A.7. Representación esquemática de una red neuronal.	108

Índice de Tablas

Tabla 2.1. Datos técnicos de los videos utilizados como bases de datos.	28
Tabla 3.1. Valores de media, desviación estándar y umbrales T_1 y T_2 para las regiones de interés de las imágenes correspondientes a los videos de motor tipo péndulo y salto de gimnasta.	35
Tabla 3.2. Funciones de Matlab utilizadas en algoritmo “edp_AI” y su descripción.	38
Tabla 3.3. Porcentajes de falsos negativos para videos de gimnasta, considerando la cantidad total de imágenes por video.	44
Tabla 3.4. Porcentajes de falsos negativos para videos de péndulo, considerando la cantidad total de imágenes por video.	44
Tabla 3.5. Porcentajes de falsos negativos para video de robot Fanuc, considerando la cantidad total de imágenes del video.	44
Tabla 3.6. Precisiones de ajuste de trayectoria para distintos valores de la tasa de aprendizaje η y diferentes estrategias de aprendizaje.	51
Tabla 3.7. Conjunto de pesos para las redes neuronales entrenadas para uno de los videos de motor y de salto de gimnasta.	53
Tabla 3.8. Ecuaciones polinomiales para una de las trayectorias seguidas por el motor y una trayectoria seguida por la gimnasta al hacer un salto.	56
Tabla 3.9. Comparación del error cuadrático medio (ECM) de los métodos de red neuronal e interpolación.	57
Tabla 3.10. Tiempos de ejecución de algoritmos en base a red neuronal e interpolación para el caso de la trayectoria de Salto A.	58
Tabla 4.1. Mediciones de iluminación y gráfica de trayectorias para distintas combinaciones de lámparas encendidas y apagadas.	62
Tabla 4.2. Error cuadrático medio al comparar trayectoria de referencia con distintos patrones de iluminación para videos de motor con movimiento de vaivén.	65
Tabla 4.3. Comparación de parámetros de similitud para distintos saltos.	77
Tabla 4.4. Parámetro de evaluación ϵ para las comparaciones entre las trayectorias de gimnasta.	80

Capítulo 1

Introducción

1.1 Planteamiento del problema

La sociedad avanza de manera constante en cuanto a desarrollo tecnológico se refiere, más aún a partir de mediados del siglo XX y hasta la fecha, periodo se ha llegado a conocer como “la era de la información” por revistas como Forbes y New York Times [1]. En este contexto, los sectores tanto industrial como académico e incluso doméstico, han sido testigos del aumento en la demanda y mejora en aspectos como: mayor rapidez de producción de un determinado artículo, más facilidad de uso de un producto, demanda del incremento en el nivel de seguridad, mejora de la eficiencia en tareas médicas, etc. que va exigiendo una sociedad cada vez más avanzada y cuya población va en aumento. Una de las ciencias que ha tenido un gran auge en fechas recientes, y que ha contribuido a resolver gran parte de los problemas como los mencionados, es la visión por computadora, así como el procesamiento y análisis de imágenes digitales. La visión por computadora puede ser definida como el área de la ciencia que consiste en la representación en imágenes digitales correspondientes a la apariencia y contorno de uno o varios objetos en tres dimensiones mediante la implementación de algoritmos matemáticos [2].

Según la empresa francesa Essilor [3] y la Asociación Americana de Optometría (AOA) [4], la vista provee el 80% de la información que recibe el ser humano de su entorno, y por lo tanto la visión es crucial en el desempeño de las tareas y de la capacidad de un individuo. El dotar a una máquina o proceso de la habilidad de la percepción visual, es una forma de automatizar tareas que son tediosas o peligrosas al ser humano, y contribuye de manera significativa a la eficiencia en la ejecución y volumen de tareas realizadas. La proliferación de computadoras cada vez más avanzadas, disponibilidad de cámaras de alta calidad y relativamente baratas, y el incremento en la necesidad de sistemas que puedan llevar a cabo análisis de imágenes y video, han generado un gran interés hacia

Herramientas para determinación de regularidad de movimiento articulado mediante el seguimiento en una secuencia de imágenes

la investigación de seguimiento de objetos [5], este proceso consiste en la estimación de la trayectoria de objetos en movimiento en una secuencia de imágenes [6]. Dentro del ámbito de la visión por computadora, el uso de seguimiento de trayectorias es pertinente en tareas como:

- Interfaces humano-computadora: Reconocimiento de gestos, seguimiento de miradas, etc. [7].
- Detección del color de piel (*Skin color detection*) [8]
- Comunicación por video [9].
- Aplicaciones de sistemas de seguridad y vigilancia [10].
- Sistemas de realidad aumentada [11].
- Asistencia de manejo [12].
- Casas inteligentes [13] [14].
- Análisis y clasificación de trayectorias de animales [15].
- Análisis de trayectorias de huracanes [16].
- Tráfico inteligente [17].
- Entretenimiento [18].

Estas y muchas otras actividades son una muestra del potencial que la visión por computadora, y, en particular, el seguimiento de trayectorias de objetos pueden tener. Muchas de estas aplicaciones tienen algo en común, y es que en ellas es de interés verificar la regularidad de un procedimiento repetitivo para determinar un funcionamiento correcto o un movimiento adecuado. A través de la captura en video de objetos en movimiento y el análisis de sus trayectorias es posible obtener una descripción espacio-temporal de un objeto para clasificación de su actividad, análisis de su comportamiento, o descripción de su interacción con el entorno [19] Sin embargo, aún con todos los avances y aplicaciones desarrolladas, todavía se tienen bastantes problemas a resolver en cuanto a obtención de trayectorias se refiere; uno de ellos es que, a mayor cantidad y calidad de herramientas de análisis de video, también es mayor la cantidad de datos que se tienen y por lo tanto se incrementa el requerimiento computacional para su procesamiento. Otro problema a tener en cuenta es que en muchas aplicaciones, el seguimiento y obtención de trayectorias, así

como el análisis de estas, aún se realiza en forma manual, ocasionando que la tarea sea laboriosa y tediosa, sin mencionar las repercusiones que puede ocasionar el dejar estos análisis al criterio y capacidad humanas, el ser humano es un ente proclive a tener equivocaciones, más aún al realizar una tarea larga y tediosa, además de ser propenso a sufrir enfermedades o simplemente cansancio que pueden ocasionar subjetividades que afecten los resultados de los análisis [20].

Las investigaciones e implementaciones realizadas en este trabajo vienen a solucionar una parte de los problemas mencionados, en primer lugar, en cuanto a exceso de datos se refiere, se brinda una forma de reducir datos de video a funciones matemáticas comparativas cuya información sea incluso más relevante (sin ruido) que los datos de video tenidos en un inicio, y con mucho menor requerimiento en demanda de costo computacional. Las técnicas y algoritmos presentados en este trabajo también vienen a reducir el tiempo y la dificultad de obtención y análisis de trayectorias comparado con los métodos manuales utilizados en la actualidad, esto mediante herramientas que brindan facilidad de uso y ofrecen una gama de parámetros que ayuden en la comparación de trayectorias. Sin más preámbulos, se puede decir que si bien lo que aquí se ofrece no es una solución general a todos los problemas de obtención y análisis de trayectorias, sí se plantea una herramienta que viene a complementar lo que se está haciendo hoy en día en materia de este tema, facilitando en gran medida las tareas a realizar y buscando tener mejores resultados.

Herramientas para determinación de regularidad de movimiento articulado mediante el seguimiento en una secuencia de imágenes

1.2 Objetivos

Los objetivos generales y particulares del presente trabajo son mencionados a continuación:

1.2.1 Objetivos Generales

Desarrollar algoritmos para el seguimiento y caracterización de trayectorias de objetos, que formen la base de una herramienta para comparación de distintas trayectorias, así como analizar su afectación por cambios de la iluminación.

1.2.2 Objetivos particulares

Obtener la trayectoria de objetos rígidos articulados en movimiento a partir de una secuencia de imágenes adquiridas de un video.

Caracterizar el comportamiento de las trayectorias generadas a partir del movimiento realizado por objetos determinados, pero reduciendo la cantidad de imágenes de entrada.

Establecer parámetros que aporten información para la caracterización de trayectorias seguidas por objetos.

Obtener información acerca de cómo el seguimiento de trayectorias de objetos a partir de imágenes se ve afectado por cambios de iluminación.

Comparar distintas trayectorias y determinar diferencias o irregularidades de unas con respecto a las otras.

1.3 Justificación

La determinación de regularidad de movimiento de un objeto (o persona) juega un papel muy importante en el análisis de trayectorias en áreas que van desde el ámbito deportivo hasta las aplicaciones médicas, por ejemplo, el saber la trayectoria de un clavadista o una gimnasta al hacer su ejercicio de rutina determina en gran parte su puntuación en un concurso, o incluso mediante el conocimiento de trayectorias de un deportista, se pueden adquirir ideas de cómo mejorar sus movimientos en sus siguientes ejecuciones [21], de igual manera, el comportamiento del movimiento del brazo de una persona puede indicar si dicha persona está sana o está lesionada [22]. Es por esto que la presente investigación se enfoca en servir como herramienta que aporte parámetros para indicar si una trayectoria seguida por un objeto es regular, teniendo otra u otras trayectorias como referencia.

El desarrollo de la investigación aquí presentada aporta cuatro ventajas principales con respecto a otros sistemas utilizados para propósitos similares. La primera de ellas es facilidad de uso, las herramientas aquí diseñadas son muy sencillas de utilizar, de manera que se necesitan muy pocos conocimientos técnicos para poder obtener las características de trayectorias y hacer comparación entre ellas, puesto que casi todos los algoritmos trabajan automáticamente. Como segunda ventaja se tiene el costo de implementación, a diferencia de muchos sistemas en el mercado, el costo del hardware necesario para la implementación de las herramientas aquí diseñadas es muy bajo, aunque la calidad de los resultados obtenidos dependerá del hardware utilizado. Aún con la facilidad de uso y el costo bajo de implementación se han logrado resultados de detección de trayectorias fiables, con un error de detección de ± 1.3048 píxeles en el eje X y de ± 1.95 píxeles en el eje Y. Por último, las herramientas aquí investigadas engloban tres partes esenciales para la comparación y determinación de regularidad de trayectorias: Obtención de trayectorias, caracterización de trayectorias y comparación de trayectorias. Este aspecto de englobar estas tres partes esenciales en un solo sistema no se ha encontrado en ninguna otra aplicación realizada hasta el momento. Todas estas ventajas hacen que la presente investigación sea muy justificable de desarrollar.

Herramientas para determinación de regularidad de movimiento articulado mediante el seguimiento en una secuencia de imágenes

1.4 Hipótesis

“La regularidad de movimiento de un objeto puede ser determinado mediante el análisis y caracterización de parámetros de su trayectoria a partir de una secuencia de imágenes, aun con reducción de la tasa de video”.

1.5 Alcances

Crear un conjunto de herramientas algorítmicas que sirvan como ayuda en la determinación de la regularidad de una trayectoria seguida por uno o varios objetos en movimiento, aportando parámetros de comparación y selección de sub-muestro del video para la caracterización de dicha trayectoria. Este conjunto de herramientas se encarga de obtener trayectorias a partir de secuencias de videos, adquirir características de estas trayectorias y comparar las características de distintas trayectorias para analizar la regularidad de unas con respecto a otras. Los objetos a analizar deben ser objetos sólidos, pudiendo ser uno o varios objetos, siempre y cuando estos no se crucen entre sí al realizar sus movimientos (que no haya cruce de trayectorias). Las herramientas pueden funcionar con baja calidad fotométrica, utilizando cámaras de video convencionales y/o técnicas de sub-muestreo de video, aun así se puede lograr un reconocimiento de la posición promedio de los objetos por medio de operaciones simples.

1.6 Limitaciones

Debido a que la definición los parámetros de comparación para determinar regularidad de una trayectoria dependen de la aplicación, es difícil llevar a cabo la investigación con propósitos universales.

Bases de datos (videos) estándar difíciles de adquirir para esta aplicación en particular, por lo que se tienen una cantidad limitada.

Herramientas para determinación de regularidad de movimiento articulado mediante el seguimiento en una secuencia de imágenes

1.7 Metodología

La metodología consistió en un inicio en la investigación a fondo acerca de los temas de visión artificial y seguimiento de trayectoria de objetos, repaso de trabajos anteriores a esta tesis y toma de cursos necesarios para implementar la investigación.

Al mismo tiempo se fue investigando sobre los diferentes tipos de programas informáticos que podrían ser de ayuda (En especial Matlab y Adobe Photoshop), esto para la obtención de conocimientos acerca de cómo utilizar dichos programas en el área de visión artificial.

Se desarrollaron funciones en las plataformas de software mencionadas para la detección de objetos y determinación de los puntos de la trayectoria.

Posteriormente se realizó un video prototipo inicial para pruebas, y se analizaron los parámetros que afectan los resultados de obtención de trayectorias para obtener cuales deben ser las condiciones óptimas para la implementación de los algoritmos diseñados.

Una vez teniendo los conocimientos necesarios y las trayectorias de los objetos de estudio, se procedió a llevar a cabo la caracterización de las trayectorias seguidas por estos, para así obtener las ecuaciones o funciones matemáticas que los describan.

Después de haber obtenido las funciones matemáticas descriptivas de las trayectorias analizadas, se investigó acerca de los parámetros de comparación que pudieran servir para comparar una trayectoria con otras, algunos de estos parámetros fueron extraídos de fuentes como artículos, mientras que otros fueron propuestos.

Durante todo el proceso se fue redactando el documento final de tesis, escribiendo cada observación, resultado y/o conclusión determinada a partir de la experimentación.

1.8 Estructura de la tesis

En el **capítulo 1** se dejan en claro el planteamiento del problema a resolver, la justificación, los objetivos tanto generales como particulares, la hipótesis planteada al inicio de la investigación, la metodología que se siguió.

En el **capítulo 2** se presenta el estado del arte, dividiendo éste en dos secciones: Tecnologías utilizadas y algoritmos creados. Se plantea el marco teórico, los conceptos necesarios para el entendimiento pleno de los capítulos subsecuentes. Además se explican los trabajos anteriores de los cuales esta tesis sirve como complemento, también se incluye un diagrama de todos los pasos realizados en el proceso de obtención, caracterización y comparación de trayectorias, ilustrando de forma general todo este proceso, para preparar al lector para los siguientes capítulos.

El **capítulo 3** constituye una explicación detallada de los algoritmos de análisis de imágenes y obtención de trayectorias utilizados, este capítulo y los siguientes se abordan principalmente desde un punto de vista práctico. En este capítulo también se describen dos métodos utilizados para el ajuste de las trayectorias obtenidas en esta tesis: Red neuronal e interpolación por mínimos cuadrados. Se hace también una comparación de los resultados obtenidos por cada uno de estos métodos.

En el **capítulo 4** se describe un experimento realizado para determinar si la iluminación puede causar errores o interferencias en la obtención de trayectorias por medio de técnicas que utilizan el color como parámetro de detección. Este capítulo también aborda una explicación minuciosa de todos los parámetros de comparación de similitud de trayectorias para su análisis y la determinación de su regularidad, también se explica el cómo fueron obtenidos cada uno de estos parámetros mediante los algoritmos diseñados, por último se hace una muestra de los resultados conseguidos al hacer una comparación de los videos que se utilizaron como base de datos.

En el **capítulo 5**, se hacen comentarios y discusiones sobre los resultados obtenidos, así como las conclusiones determinadas a lo largo de la investigación, explicando las ventajas, desventajas y novedades que este trabajo aporta en este ámbito de estudio de

Herramientas para determinación de regularidad de movimiento articulado mediante el seguimiento en una secuencia de imágenes

trayectorias. Al final del capítulo se plantean los trabajos futuros a desarrollar para la optimización de lo aquí presentado, así como para su implementación en distintas aplicaciones.

En el **apéndice A** se muestra un glosario de términos utilizados a lo largo de la tesis, definiendo conceptos de suma importancia para el entendimiento de la misma, y dando una breve explicación de algunos de estos conceptos.

En el **apéndice B** se presentan todos los algoritmos creados a lo largo de la investigación y que son de suficiente importancia para ser presentados, de manera que el lector tenga una referencia si se quiere hacer una implementación de estos algoritmos.

En el **apéndice C** se exponen las tablas de coordenadas de las trayectorias obtenidas de los videos utilizados a modo de base de datos para trabajos futuros.

Capítulo 2

Estado del arte, marco teórico y antecedentes

En este capítulo se tratan las pautas necesarias para un mejor entendimiento de la tesis y los temas incluidos en esta; se aborda el estado del arte, así como conceptos, explicaciones clave, y los antecedentes a modo de resumen de los trabajos realizados anteriormente en la Universidad Autónoma de Baja California relacionados con la presente investigación. Al final del capítulo se muestra un diagrama de flujo que brinda una explicación general del proceso llevado a cabo a lo largo de toda la investigación.

El interés por los métodos de procesamiento de imágenes digitales se fundamenta en dos áreas principales de aplicación: a) mejora de la calidad para la interpretación humana; b) procesamiento de los datos de la escena para dotar de percepción a las máquinas de forma autónoma. En el intento de dotar a las máquinas de un sistema de visión aparece el concepto de Visión Artificial [23].

Herramientas para determinación de regularidad de movimiento articulado mediante el seguimiento en una secuencia de imágenes

2.1 Estado del arte

A continuación se trata el estado del arte concerniente al seguimiento de objetos y obtención de trayectorias. A pesar de encontrar relativamente pocas investigaciones acerca del planteamiento y uso de parámetros de similitud para comparación entre trayectorias, se han logrado recopilar las investigaciones más trascendentales relacionadas con el tema. Esta sección se organiza en dos partes, la primera de las cuales se enfoca en las tecnologías utilizadas en este momento relacionadas con el tema de la presente investigación, en cuanto a la segunda parte, esta aborda los algoritmos, técnicas y medidas utilizados actualmente para la comparación de trayectorias.

2.1.1. Estado del arte en tecnología

Existen numerosos dispositivos utilizados en el seguimiento de trayectorias de objetos, los cuales tienen aplicaciones tan diversas como entretenimiento, seguridad y en el campo de la medicina, en donde se utilizan mecanismos de detección tales como sensores de ángulo, marcas o *targets* conectados al sistema de detección, tecnología basada en láser o luz infrarroja, entre otros. Por ejemplo, en el área del entretenimiento se tienen dispositivos como el Kinect de Microsoft, el cual tiene la capacidad de detectar objetos en tres dimensiones, este producto (en su uso original, sin haber sido reprogramado), tiene un “pre-conocimiento” de la forma del cuerpo humano, por lo que puede reconocer la forma del mismo y detectar sus movimientos hasta 30 veces por segundo [18].

Existen también cámaras especializadas en la captura de movimiento para diversas áreas como la robótica, la metrología y la animación; por ejemplo la cámara VICON, estas cámaras funcionan con la necesidad de *targets*, en tiempo real y con alta resolución [24]. Otro tipo de dispositivos que son usados para las mismas áreas de aplicación que las VICON, pero que también tienen aplicaciones en el área de la medicina, es el OPTOTRAK Certus, el cual es un sistema de captura de movimiento en 3 dimensiones, este sistema

tiene, entre otros atributos, la capacidad de seguir hasta 512 *targets* al mismo tiempo a una frecuencia de captura de 4200Hz, y entrega las coordenadas de los objetivos seguidos, pudiendo crear trayectorias de los mismo. Las curvas que obtiene se pueden graficar en 2 o 3 dimensiones para su posterior análisis [25]. Sin embargo, estos dos sistemas también presentan algunas desventajas; por ejemplo, necesitan de *targets* especiales (vendidos por el mismo fabricante) para llevar a cabo el seguimiento, otra desventaja es el precio elevado de este tipo de equipo, además de su pobre índice de portabilidad.

La parte de la generación de trayectorias que se desea llevar a cabo en el presente trabajo para determinación de regularidad podría ser realizada por medio de estos tres y algunos otros sistemas vendidos en el mercado en la actualidad, sin embargo, el método aquí implementado tiene algunas ventajas sobre los demás, las cuales se discuten a continuación:

- Mayor versatilidad y portabilidad: Es decir, mayor capacidad de implementación en cualquier lugar sin tener que trasladar un equipo especial, ya que se utilizan cámaras convencionales.
- Menor Costo: Los sistemas existentes en el mercado hasta la fecha son bastante costosos, además de que requieren de equipo adicional del mismo fabricante para cada tipo de uso, mientras que el sistema aquí investigado tiene un costo relativamente bajo comparado con los de uso comercial, principalmente debido al tipo de hardware que requiere para su implementación.
- Una ventaja más es que, al ser un sistema especializado, diseñado específicamente para seguimiento de trayectorias de objetos y determinación de irregularidades de movimiento mediante comparación entre ellas, creemos que el trabajo aquí presentado tiene un buen grado de robustez para este tipo de tareas en concreto.

Herramientas para determinación de regularidad de movimiento articulado mediante el seguimiento en una secuencia de imágenes

2.1.2. Estado del arte en medición de regularidad de movimiento y comparación de trayectorias

En esta sección se abordarán los documentos y trabajos más trascendentales publicados recientemente en relación con la extracción de parámetros de trayectorias, su clasificación y comparación de similitud. En [22], se trata un método de detección de irregularidades, comparación y clasificación de trayectorias de movimientos repetitivos de los brazos de pacientes lesionados o con parálisis cerebral; tomándose en cuenta algunos parámetros como curvatura y *jittering*, término que se refiere a la vibración que tienen los pacientes en sus movimientos como resultado de la parálisis cerebral o lesión, los movimientos realizados consisten en mover un brazo desde el regazo a la boca un número determinado de veces, graficar este movimiento y de acuerdo a varios parámetros como los mencionados, clasificar estos movimientos. Para la asignación de las clases se toma como referencia una de las formas de evaluación más utilizadas en Japón para determinar el grado de daño en pacientes con algún tipo de lesión o parálisis, llamado *SIAS score (Stroke Impairment Assessment Set)* [26]. El SIAS evalúa el grado de lesión o parálisis del paciente de manera manual por un terapeuta especializado y lo agrupa en 5 niveles, donde el nivel más alto corresponde a una persona sana y el nivel más bajo corresponde a una persona con pérdida total de movimiento en sus extremidades, mediante un algoritmo de clasificación se lograron obtener resultados congruentes entre las clases asignadas por los terapeutas y las predichas por el algoritmo, de esta manera se puede comparar los resultados de máquinas con los de especialistas. Para graficar las trayectorias del movimiento de brazo de los pacientes se utiliza el sistema OPTOTRAK Certus.

En [27] se aborda un algoritmo de comparación de similitud de movimiento de diferentes tipos de objetos: automóvil, motocicleta, bicicleta y peatones, estos objetos se toman como clases para la comparación. Para llevar a cabo los análisis se obtuvieron datos de openstreetmap.org, página web que incluye una enorme base de datos de trayectorias capturadas por distintos usuarios mediante GPS; a partir de estas trayectorias se extraen ciertos parámetros los cuales se dividen en similitud espacial, temporal y espacio-temporal,

a su vez estos parámetros se someten a un algoritmo de clasificación SVM (*Support vector machine*), obteniendo precisiones de hasta 82% cuando se utilizan las cuatro clases mencionadas.

En [15], por su parte, se proponen distintas medidas o parámetros de caracterización de trayectorias seguidas por animales para su clasificación, entre estas variables se encuentran rectitud (*ST*), desplazamiento cuadrático medio (*MSD*), y sinuosidad (*SI*), el artículo se aborda solo desde un punto de vista teórico, y no se hace ningún tipo de experimentación para verificar si los parámetros propuestos son efectivos para la distinción entre trayectorias. En [21] se analiza la similitud de movimientos en karatekas, realizando una serie de patadas del mismo tipo y obteniendo diversos ángulos para este movimiento repetitivo, en este caso se utilizaron objetivos especiales, los cuales fueron ubicados en puntos estratégicos del cuerpo de los deportistas para realizar el seguimiento. La diversidad de aplicaciones mencionadas en esta sección genera una idea de todo lo que se puede lograr con este tipo de implementaciones, las cuales pueden ser mejoradas si se consideran la portabilidad y costo, buscando a la vez no sacrificar la exactitud para que la aplicación de las herramientas sea de utilidad.

2.2 Representación de imágenes digitales

Una imagen digital puede ser definida como una representación matricial de dimensiones $M \times N$, donde M representa las filas y N las columnas de la imagen. El contenido de esta matriz representa la intensidad de cada pixel en una ubicación (X,Y) de la imagen. La figura 2.1 muestra un ejemplo de una representación matricial para una imagen digital en escala de grises, la matriz mostrada (lado derecho de la figura), representa los valores de intensidad de los pixeles marcados por el cuadro blanco en la imagen (lado izquierdo de la figura); existe una diversidad de nomenclaturas para la representación de matrices y pixeles correspondientes a una imagen, la que aquí se utiliza es la mostrada en la figura 2.1, donde X e Y son las coordenadas espaciales y el valor $f(X,Y)$ en cualquier punto es la intensidad o nivel de gris en la imagen en ese punto [23].

Herramientas para determinación de regularidad de movimiento articulado mediante el seguimiento en una secuencia de imágenes

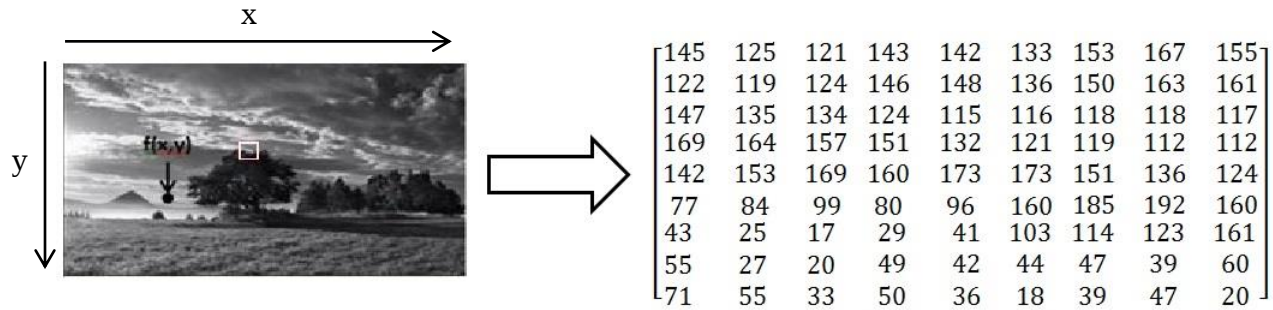


Figura 2.1. Representación matricial de una imagen digital en escala de grises. La matriz del lado derecho está dada por el valor de intensidad de los píxeles en el recuadro blanco que está entre la parte superior del árbol y el cielo de la imagen del lado izquierdo.

Al igual que las imágenes digitales (las cuales a partir de este punto se les llamará simplemente imágenes) en nivel de gris, las imágenes a color RGB (*Red*, *Green*, *Blue*) pueden ser representadas por medio de arreglos matriciales, con la diferencia de que, en lugar de ser una sola matriz para toda la imagen, será una matriz por cada uno de los tres colores RGB. Como se muestra en la figura 2.2, los cuadros negros mostrados en la imagen (lado izquierdo) son regiones de 3x3 píxeles, cada una de estas regiones está dada por tres matrices $R(x,y)$, $G(x,y)$ y $B(x,y)$, las cuales representan los tres colores primarios utilizados en computación: rojo, verde y azul, cualquier combinación de estos tres colores dará como resultado un color distinto. Por ejemplo, suponiendo que cada color de la figura 2.2 tiene su máxima intensidad (255) se puede observar en el lado derecho de la figura 2.2 que si la región marcada por un cuadro negro se encuentra dentro del color rojo, la matriz $R(x,y)$ tendrá solo valores de su más alta intensidad, es decir, 255, mientras que los valores de las matrices $G(x,y)$ y $B(x,y)$ serán de 0, si la región marcada por el cuadro negro se encuentra en el centro de la imagen del lado izquierdo, se tendrán los mayores valores de intensidad posibles para las tres matrices de colores $R(x,y)$, $G(x,y)$ y $B(x,y)$, y de la combinación de estas tres matrices se obtendrá el color blanco.

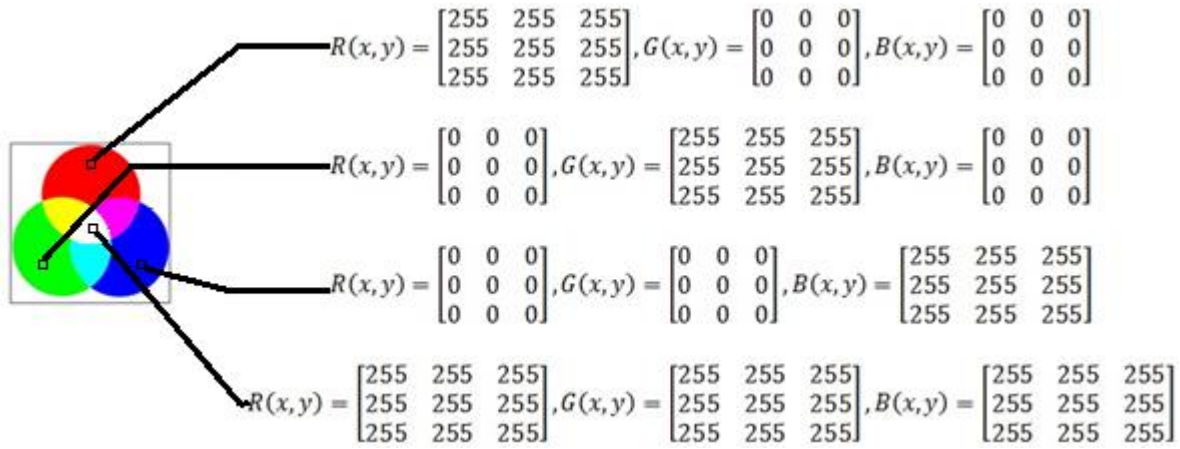


Figura 2.2. Representación matricial de una imagen a color RGB, los cuadros negros del lado izquierdo de la imagen son regiones de 3x3 pixeles, y cada región está dada por tres matrices $R(X,Y)$, $G(X,Y)$, $B(X,Y)$.

La obtención de la intensidad por pixel para imágenes RGB se lleva a cabo de la misma manera que para las imágenes en escala de grises, con la excepción de que para las imágenes RGB existirán tres valores de intensidad asociados a cada pixel (uno por cada color), estos tres valores darán como resultado la intensidad de un determinado pixel. Un tercer tipo de imagen, cuya explicación es necesaria para el desarrollo del presente trabajo, son las imágenes binarias, estas son imágenes cuya matriz correspondiente solo puede contener valores de intensidad de 0's y 1's, donde los 0's representan la intensidad más baja y se aprecian en color negro y los 1's representan la intensidad más alta y se aprecian como color blanco, un ejemplo de este tipo de imágenes se puede observar en la figura 2.3, si cada cuadro de la imagen (lado izquierdo) se toma como un pixel, la matriz correspondiente a esa imagen quedaría como se observa (lado derecho).

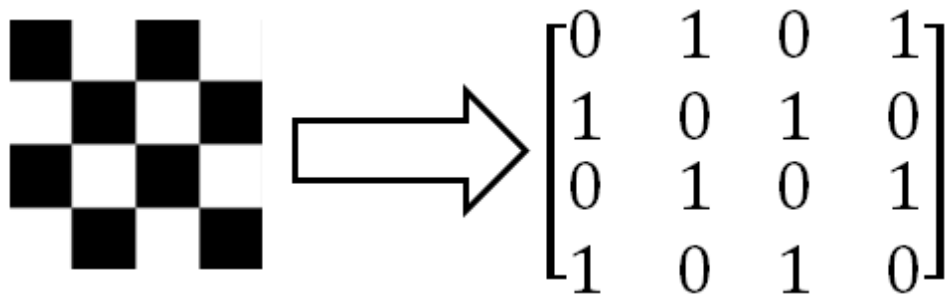


Figura 2.3. Representación matricial de una imagen binaria, cada cuadro de la imagen del lado izquierdo es tomado como un pixel.

2.3 Selección de características para seguimiento de objetos

En visión por computadora, existen tres pasos clave para el seguimiento y caracterización de trayectorias de objetos: en primer lugar está la detección del objeto u objetos de interés, continuando con el seguimiento de la trayectoria de dichos objetos de cuadro a cuadro, por último se tiene análisis de la trayectoria para determinar el comportamiento del objeto. Sin embargo, llevar a cabo un seguimiento óptimo puede ser un problema debido a diversos factores que afectan sus resultados, estos factores pueden deberse a [28]:

- Pérdida de información causada por la proyección de un mundo en 3D a una imagen 2D.
- Ruido en las imágenes.
- Movimientos complejos por parte de los objetos.
- Objetos no rígidos (por ejemplo agua).
- Oclusiones parciales o totales en objetos.
- Formas complejas.

Para simplificar estos problemas, se deben imponer limitaciones al movimiento o apariencia de los objetos a seguir, por ejemplo, movimientos más suaves y/o de velocidad constante. Otra forma de reducir estos problemas es conocer de antemano el número de objetos a seguir, así como su apariencia o forma. La selección de las características correctas para llevar a cabo el seguimiento es una tarea crítica [28]. En general, se busca que las características de un objeto determinado sean únicas, para que puedan ser distinguidas de otros objetos, o si son varios objetos con las mismas características, lo ideal es que las trayectorias de estos no se crucen. Dependiendo de la aplicación y de los objetos que se deseen seguir, son las características que se deberán tener en cuenta, a continuación se enlistan las más comunes:

1. **Color:** El color aparente de un objeto está influenciado por dos factores físicos. Primero se tiene la distribución espectral de la iluminación, y en segundo lugar las

propiedades reflectoras del objeto. En procesamiento de imágenes, una forma de representar toda la gama de colores es mediante el modelo RGB (Rojo, verde y azul), aunque las diferencias entre los colores en el espacio RGB no corresponden a las diferencias de colores percibidas por los humanos [29].

2. **Bordes:** Los límites de los objetos usualmente generan cambios fuertemente perceptibles en la intensidad de las imágenes. La detección de bordes es utilizada para percibir estos cambios. Una propiedad importante de los bordes, es que son menos sensibles a los cambios de iluminación comparados con los colores [30].
3. **Flujo óptico:** está representado por un campo de vectores que definen la traslación de cada pixel en una región. Es obtenido utilizando “restricción de brillo”, debido a que se asume la constancia de brillo de pixeles correspondientes a dos cuadros consecutivos de imágenes [31].
4. **Textura:** La textura es una medida de la variación de la intensidad de una superficie, que cuantifica su suavidad y regularidad. Al igual que las características de bordes, la textura es menos sensible a la iluminación comparada con el color [32].

En el caso de la presente investigación, la característica que se ha seleccionado para llevar a cabo la detección de objetos es la de color, esto debido a que brindó resultados aceptables en el trabajo utilizado como base para esta investigación [33], además facilita la implementación de los algoritmos diseñados, por ejemplo, los colores de un objeto son relativamente fáciles de detectar en comparación con el brillo de pixeles de imágenes consecutivas o el borde de un objeto no uniforme como una persona. También se podría decir que la detección se efectúa mediante una combinación entre características de color y bordes, puesto que las imágenes se pasan por un proceso de segmentación, y en este punto se utilizan los bordes de los objetos para detectar sus centroides.

2.4 Detección de objetos

Como mecanismo de detección de objetos para realizar el presente trabajo, se eligió la segmentación, (Ver Apéndice 0 para observar distintos mecanismos de detección de objetos), este mecanismo consiste en dos partes, umbralado para detección de objetos y etiquetado de los objetos detectados, a continuación se explican ambas partes.

2.4.1 Umbralado de una imagen para detección de regiones ocupadas por un objeto.

Para llevar a cabo el método de segmentación y extraer una región de interés en una imagen, ya sea en escala de grises o a color, se debe aplicar un proceso al cual se le conoce como umbralado, esta tarea consiste en buscar las intensidades de los pixeles de la región de interés; una vez conocidas estas intensidades, se aplica el criterio de la ecuación (2.1) para crear una imagen binaria [2].

$$g(x, y) = \begin{cases} 0 & \text{si } f(x, y) > T \\ 1 & \text{si } f(x, y) \leq T \end{cases} \quad (2.1)$$

Donde $g(x,y)$ será el valor del pixel resultante después de aplicar un umbral T a la al pixel de la imagen original, cuyo valor de intensidad era $f(x,y)$. El resultado será un conjunto de pixeles con valores de 0's y 1's, los cuales formarán una imagen binaria. El encontrar un umbral adecuado para una imagen determinada es una tarea muy complicada si se desea hacerlo a prueba y error, sin embargo, esto se puede facilitar considerablemente si se utiliza la información proporcionada por el histograma de la imagen, así como las propiedades estadísticas del mismo (Ver Apéndice B para obtener una definición de histograma y sus propiedades estadísticas). Por ejemplo, para encontrar el umbral adecuado para la figura 2.4, el umbral se eligió al restar y sumar la desviación estándar de la media del histograma (ecuaciones (2.2) (2.3)).

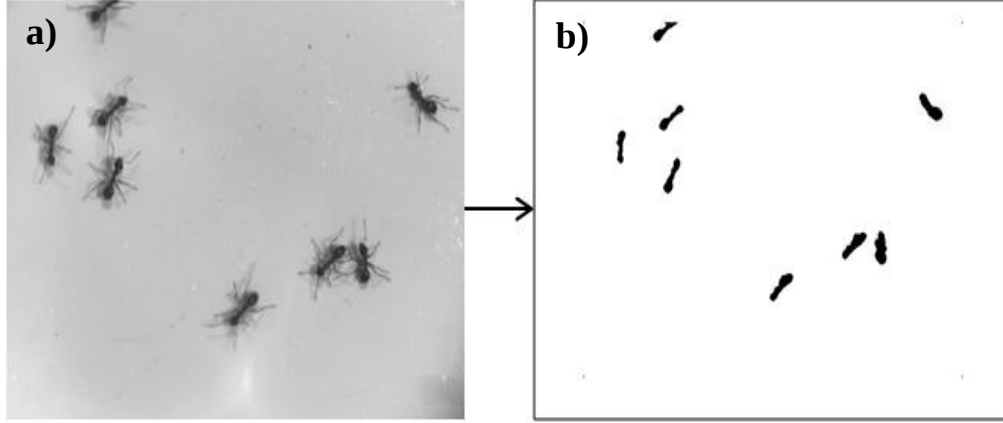


Figura 2.4. Segmentación de una imagen por medio de detección de umbral. a) Imagen original, b) Imagen resultante después de haber aplicado el método de segmentación con asignación de umbral (Imágenes extraídas de Q. Sánchez, "Estimación de Movimiento de Objetos a Partir de una Secuencia de Imágenes", [33]).

$$T_1 = \bar{g} - \sigma \quad (2.2)$$

$$T_2 = \bar{g} + \sigma \quad (2.3)$$

De esta manera no solo se impone un límite de umbral como en la ecuación (2.1), sino que se establece un rango de valores de intensidades como muestra la ecuación (2.4).

$$g(x, y) = \begin{cases} 0 & \text{si } T_1 > f(x, y) > T_2 \\ 1 & \text{si } T_1 \leq f(x, y) \leq T_2 \end{cases} \quad (2.4)$$

Si el valor de intensidad de un determinado pixel $f(x, y)$ se encuentra dentro del rango de umbrales T_1 y T_2 , se le establece un valor de 1, mientras que si la intensidad de un pixel $f(x, y)$ está fuera este rango, se le asigna un valor de 0, dando como resultado la imagen binaria de la figura 2.4(b). Distinguiendo a los pixeles del objeto de lo que pudiera ser el fondo o saturación de iluminación.

Herramientas para determinación de regularidad de movimiento articulado mediante el seguimiento en una secuencia de imágenes

2.4.2 Etiquetado de objetos detectados

Una vez detectadas las regiones pertenecientes al objeto u objetos de interés, se debe encontrar sus centroides, el etiquetado de objetos consiste en la detección de un centroide en cada región de interés detectada mediante la técnica de umbralado, una vez obtenido este centroide, se le asigna una etiqueta que lo distinga de los demás centroides que aparezcan en la misma imagen, en la figura 2.5 se muestra la figura 2.4 después de habersele aplicado el proceso de etiquetado. En la sección 3.3 de este trabajo se puede ver la implementación y descripción de este algoritmo.

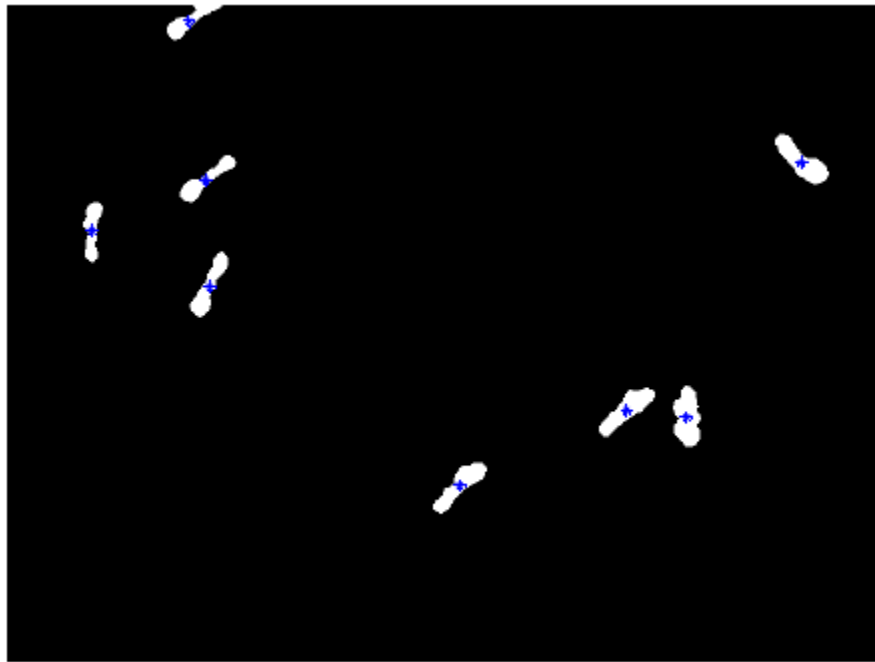


Figura 2.5. Resultado de la figura 2.4 después de haber aplicado proceso de etiquetado.

2.5 Ajuste de curvas

Cuando se trata de encontrar una curva que represente la unión o comportamiento de un conjunto de puntos, se pueden utilizar técnicas matemáticas que ejecuten esta tarea, a la cual se le conoce como ajuste de curvas, las técnicas utilizadas en esta investigación se describen a continuación.

2.5.1. Redes Neuronales

La aplicación del presente trabajo requiere la implementación de una red neuronal para un problema de regresión con el fin de determinar una curva que involucre los puntos de la trayectoria que siguió un objeto. En la figura 2.6, se tiene un conjunto de puntos que se ajustan a una curva, se tiene una función polinomial representada por la línea punteada, mientras que la curva ajustada por la red neuronal se representa por la línea continua [34]. Las redes neuronales son uno de los métodos más utilizados para resolver problemas tanto de clasificación como de regresión, esto debido a que son capaces de ajustar o predecir modelos de datos bastante complejos. Sin embargo, una de las desventajas principales para la implementación de estos sistemas es que necesitan de un número relativamente elevado de parámetros de sintonización con respecto a otros métodos para poder implementar el proceso de aprendizaje al cual se le conoce como entrenamiento (Ver apéndice 0 para una definición más detallada de red neuronal).

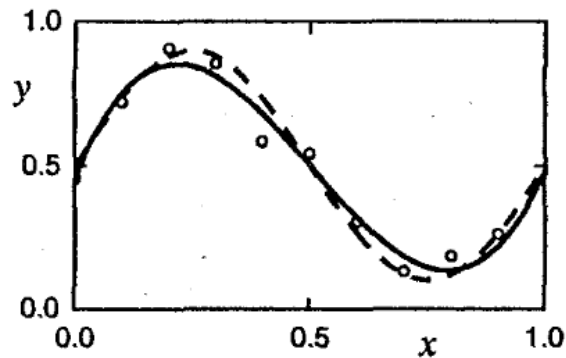


Figura 2.6. Aplicación de una red neuronal para un problema de regresión en polinomios (Imagen extraída de C. M. Bishop, “Neural Networks for Pattern Recognition”, [34]).

Herramientas para determinación de regularidad de movimiento articulado mediante el seguimiento en una secuencia de imágenes

2.5.2. Ajuste de curvas mediante mínimos cuadrados

Otra técnica de ajuste de curvas que se ha utilizado en la presente investigación consiste el método de mínimos cuadrados, este método trata de ajustar una curva que describa una serie de puntos de la mejor manera posible, minimizando el error cuadrático que existe entre los puntos y la curva aceptada en todo su trayecto, de esta manera, se puede obtener una función polinomial de grado N que describa una trayectoria de puntos (X,Y) de manera que se pueda conocer el valor de Y para cualquier nuevo valor de X, el ajuste de mínimos cuadrados también logra un resultado semejante al de la figura 2.6.

2.5.3. El error cuadrático medio

Para determinar la eficacia de los algoritmos de regresión o ajuste de curvas, es necesario utilizar herramientas matemáticas estándar de evaluación, esto para comparar cuál de los algoritmos utilizados para ajuste de curvas es mejor para esa aplicación en particular. Una de las herramientas más utilizadas es el Error Cuadrático Medio (ECM), el cual es una medida del error que existe entre la curva ajustada y los puntos para los cuales se hizo el ajuste, la ecuación para el error cuadrático medio es la de la ecuación (2.5).

$$ECM = \frac{1}{m} \sum (Y_T - Y_P)^2 \quad (2.5)$$

Donde m es el total de puntos para los cuales se ajustó la curva, Y_T son los datos verdaderos y Y_P son los datos predichos por el algoritmo de ajuste. El uso de esta medida es muy frecuente debido a que penaliza el error al elevarlo al cuadrado, lo que da una idea más clara de la diferencia en eficiencia entre dos o más algoritmos de ajuste, puesto que hace que esta diferencia sea mayormente perceptible.

2.6 Trabajo base

Este trabajo de tesis sirve de continuación al proyecto titulado “Estimación de movimiento de objetos a partir de una secuencia de imágenes” [33]. El trabajo mencionado tuvo por objetivo llevar a cabo la generación de puntos coordenados de trayectorias de objetos a partir de la división de un video en imágenes, para esto se utilizó el software Matlab. El registro de trayectorias se realizó para dos tipos de escenas diferentes, en primer lugar se registró la trayectoria seguida por un grupo de hormigas (ver figura 2.7(a)), y en segundo lugar para los ejes de movimiento de un brazo robótico marca GE-Fanuc LR MATE 200i (ver figura 2.7(b)). Las gráficas de la trayectoria seguida por los objetos se observan en la figura 2.8. Los algoritmos realizados en el trabajo de [33] fueron optimizados para la presente investigación en cuanto a tiempo de ejecución y líneas de código, los algoritmos optimizados se muestran en los apéndices B.1 y B.2.

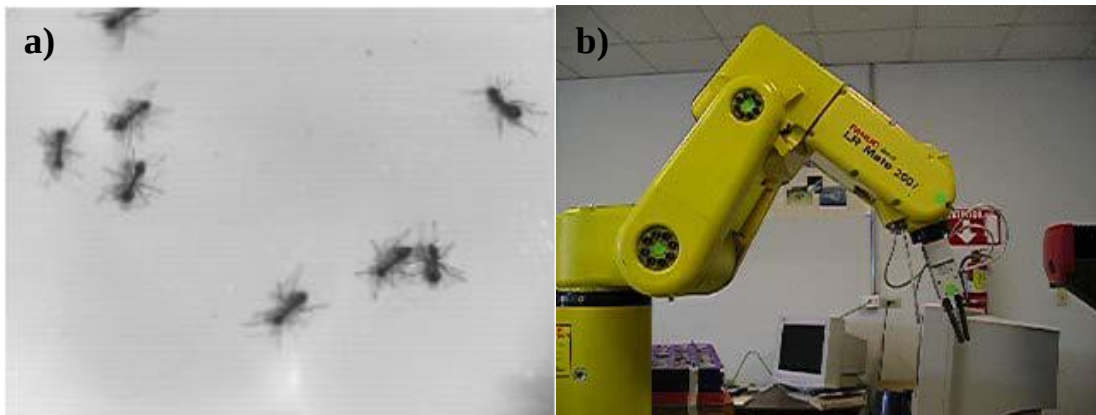


Figura 2.7. Objetos seguidos en el trabajo de tesis de Juan Carlos Quiroz [33], (a) conjunto de hormigas, (b) robot GE-Fanuc LR MATE 200i.

Herramientas para determinación de regularidad de movimiento articulado mediante el seguimiento en una secuencia de imágenes

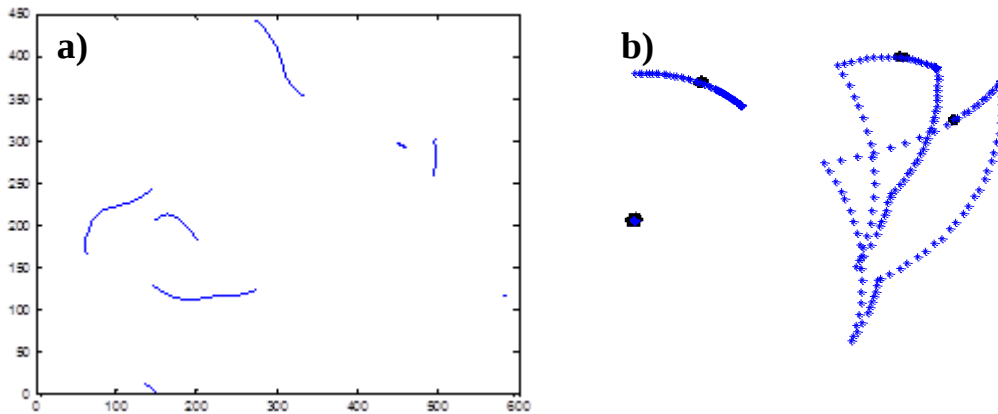


Figura 2.8. Resultados gráficos de la trayectoria seguida por un objeto, (a) Grupo de hormigas, (b) Brazo robótico GE-Fanuc LR MATE 200i [33].

El seguimiento de la trayectoria a la serie de imágenes se llevó a cabo por medio de la técnica de segmentación. El resultado de aplicar esta técnica fue un conjunto de imágenes binarias de los objetos contrastados con el fondo de la imagen (objetos en blanco y fondo negro), y a partir de estas imágenes binarias se obtienen los objetos que existen en ellas (se etiquetan regiones compuestas por grupos de píxeles vecinos). Después se referenció el centroide de cada objeto de una imagen con respecto al de la siguiente para comprobar que seguían criterios de trayectoria. El resultado final fue un método capaz de graficar las trayectorias de los objetos de la figura 2.7. Con base en este trabajo, se dio a la tarea de hacer una extensión de la información recopilada del seguimiento de la trayectoria de los objetos, agregando también la caracterización del movimiento, es decir, las ecuaciones que describen la trayectoria de los objetos y su comparación para calificar el desempeño de la trayectoria.

2.7 Bases de datos

A lo largo de la presente investigación se estuvo trabajando con cuatro tipos de videos; los cuales se tomaron como base de datos para experimentación, los primeros dos videos son correspondientes a las imágenes mostradas en la figura 2.7 y se utilizaron para hacer comparaciones entre el algoritmo de [33] y los algoritmos aquí implementados, de esta manera se pueden analizar las mejoras realizadas, por su parte, el video correspondiente al robot Fanuc, también se utilizó para observar la eficiencia de los algoritmos de generación de trayectorias para videos con trayectorias cerradas y seguimiento de múltiples objetos en condiciones controladas, Los resultados de las pruebas con este video se muestran en la sección 3.4. En la figura 2.9(a), se observa un prototipo construido para pruebas y experimentación en la investigación, a este modelo se le asignaron ciertos *targets* en el centro del mecanismo, extremo del brazo y en los dos interruptores de límite de tipo rodillo ubicados a ambos lados del mecanismo para controlar su movimiento de vaivén. Para hacer análisis y experimentaciones con respecto a los cambios de iluminación en un ambiente controlado, el motor con movimiento tipo péndulo se ubicó en una mesa especial con cuatro lámparas y se le tomó un total de diez videos, estos videos se tomaron para diferentes configuraciones de lámparas encendidas y apagadas para observar cómo la iluminación afecta la percepción de la trayectoria de un objeto al ser captado por una cámara de video (Ver Capítulo 4 para una explicación más detallada).

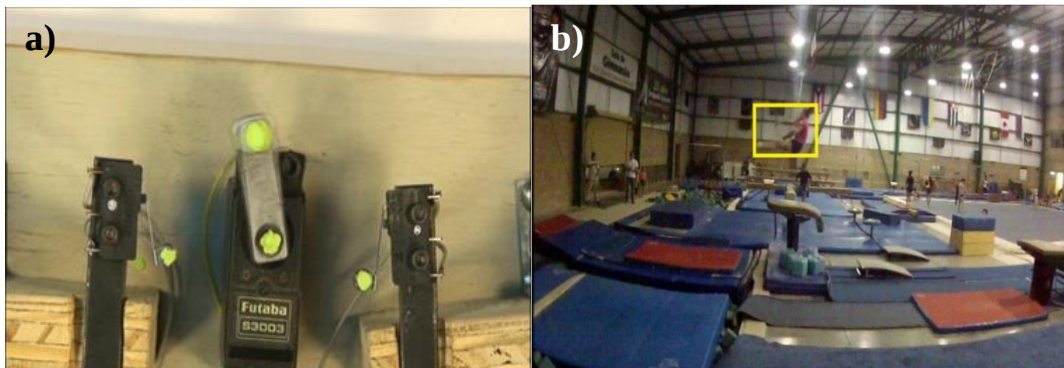


Figura 2.9. Videos utilizados como bases de datos para experimentación. (a) Motor con movimiento de vaivén tipo péndulo. (b) Gimnasta realizando un salto (ver recuadro amarillo).

Herramientas para determinación de regularidad de movimiento articulado mediante el seguimiento en una secuencia de imágenes

En la figura 2.9(b), se puede ver una imagen correspondiente a una serie de videos tomados a una gimnasta al realizar saltos de entrenamiento, estos videos también sirvieron para pruebas y experimentación, pero esta vez en un ambiente no controlado, sin *targets*, con variaciones de iluminación y una gama de colores mucho mayor que en los demás videos, con esto se pudo analizar si los algoritmos diseñados pueden funcionar para este tipo de entorno. Los videos de saltos son tres, en dos de ellos se realizó el mismo tipo de salto, mientras que en el tercero se realizó un tipo de salto diferente, se hicieron comparaciones de las trayectorias seguidas al efectuar los dos tipos de saltos. En la tabla 2.1 se muestra la información de los videos utilizados como bases de datos.

Tabla 2.1. Datos técnicos de los videos utilizados como bases de datos.

Parámetro de video	Robot FANUC	Motor de vaivén	Gimnasta
Velocidad de cuadro	15 cuadros/segundo	30 cuadros/segundo	120 cuadros/segundo
Formato de Archivo	.AVI	.WMV	.MP4
Resolución	320x240	640x480	840x480
Formato de Color	RGB24	RGB24	RGB24
Duración	28 Segundos.	10 Segundos.	Salto A: 4.12 Segundos
			Salto B: 5.22 Segundos
			Salto C: 6.12 Segundos
Tipo de cámara utilizada	Cámara digital Pentax Optio 450	Camara de video Sony handycam DCR-TRV22	Especial con mayor rapidez de toma.

2.8 Contexto de la investigación

Antes de iniciar la explicación detallada de la investigación realizada, se debe tener en cuenta el contexto de la misma, a continuación se mencionan los puntos necesarios a considerar: Todo el proceso de generación de curvas de trayectoria de los objetos se realiza fuera de línea, es decir, primero se adquieren los datos a analizar (videos) para después hacer el análisis. Se requiere una cámara fija durante toda la toma, y que los objetos de interés estén en todo momento dentro del campo de visión. Es esencial una iluminación constante dentro de un mismo video así como entre los distintos videos a comparar, esto se determina después de haber hecho una serie de pruebas, las cuales se abordan en el capítulo 5. Todos los algoritmos fueron diseñados bajo el entorno de Matlab, y se utilizó Adobe Photoshop para evaluación cualitativa.

2.9 Diagrama general del proceso total

En la siguiente página se muestra un diagrama general del proceso completo llevado a cabo en la presente investigación, presentando cada uno de los pasos tomados de una manera fácil y rápida de entender, todo este proceso se irá explicando detalladamente a lo largo de los siguientes capítulos.

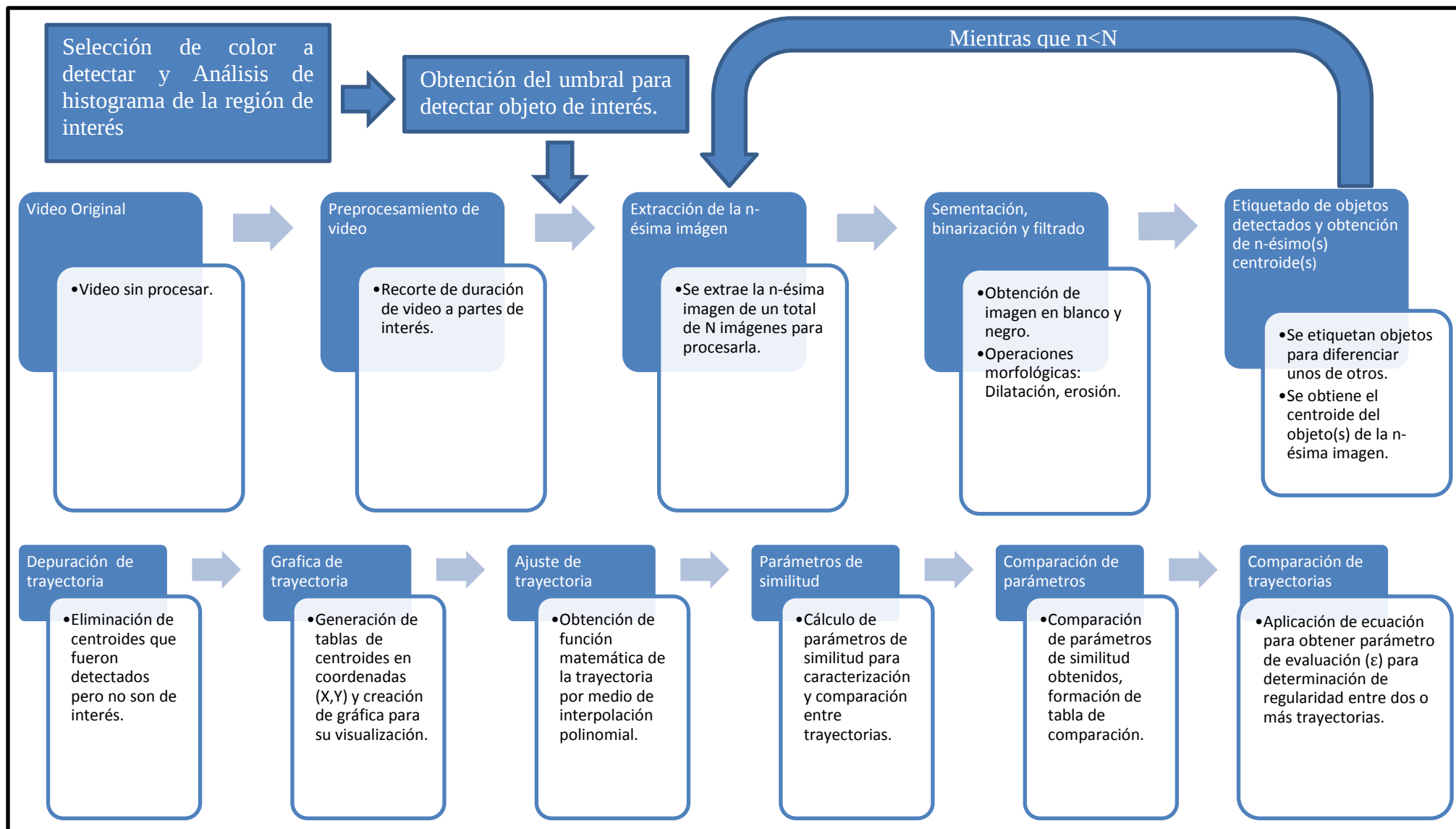


Figura 2.10. Diagrama general del proceso total de la presente tesis.

Capítulo 3

Implementación de algoritmos para obtención de trayectorias y ajuste de curvas

Con un enfoque en los videos de la presente investigación, a lo largo de este capítulo se verá cómo se llevó a cabo el proceso de obtención de trayectorias; desde el momento que se adquiere un video, el pre-procesamiento, conversión del video a imágenes, umbralado, segmentación y finalmente obtención de tabla de coordenadas en (X, Y) de los centroides de los objetos de interés detectados a lo largo de todo el video. Estos centroides forman la trayectoria seguida por el/los objeto(s) de interés.

3.1 Separación de un video en imágenes para su procesamiento

Para obtener la trayectoria o detectar el movimiento de un objeto, no basta con una sola imagen, sino que se necesita de la visualización y análisis de una serie de imágenes relacionadas secuencialmente para la detección de cambios de un cuadro con respecto al siguiente. De esta manera se establece que un determinado objeto dentro de dos o más imágenes consecutivas ha tenido un movimiento a lo largo del tiempo. El proceso inicia tomando un video y extrayendo un conjunto de imágenes a analizar de manera secuencial, a cada imagen se le asigna un índice de cuadro i , donde $i=1$ en la primera imagen extraída. El algoritmo diseñado, además de extraer imágenes del video, las analiza y obtiene los centroides de los objetos detectados en la cada imagen, formando así la trayectoria del objeto seguido. Todo esto se realiza en un mismo algoritmo, lo cual permite un ahorro de tiempo al ejecutarlo una sola vez para todo el proceso de obtención de trayectorias; este algoritmo se nombró **edp_AI** y se muestra en el apéndice B.1. El algoritmo **edp_AI** permite también establecer sub-muestreo en la extracción de imágenes del video a analizar, pudiendo reducir aún más el tiempo de ejecución, aunque con posible pérdida de resolución de la trayectoria.

3.2 Determinación de la intensidad de colores rojo, verde y azul (RGB) de una imagen en un área de interés

Una vez teniendo la división del video en imágenes, el siguiente paso a tomar para la obtención de trayectorias, es la determinación de la intensidad de color de la zona donde se ubica el objeto de interés a seguir, al trabajar con imágenes a color tipo RGB, se deberá determinar la intensidad de cada uno de los colores principales rojo, verde y azul de las regiones de la imagen que son de interés, para posteriormente establecer un umbral adecuado que contraste los objetos a seguir con respecto al fondo de la imagen; esto se puede hacer a partir del histograma de estas regiones, a continuación se explica cómo obtener este histograma.

3.2.1. Obtención del histograma de las regiones de interés

El histograma de una imagen, o de alguna región de la imagen, se puede obtener utilizando el software de edición de imágenes Adobe Photoshop. Entre las herramientas de este programa están la selección de regiones de píxeles por intensidad color o escala de grises, así como la posibilidad de observar el histograma de la imagen en general o solo de la región seleccionada, herramienta que se la aplicación aquí planteada. A continuación se describen los pasos a seguir para la selección de regiones de interés, obtención de histogramas y determinación de las intensidades de dichas regiones:

- 1) Abrir imagen a analizar.**
- 2) Selección del área de interés:** Se debe seleccionar el área de interés a analizar, es decir, el área donde se encuentra el/los objeto(s) a seguir, esto se hace con los siguientes pasos:
Menú *Select – Color range* – Seleccionar los objetos de interés – Ok.
- 3) Ver Histograma de la imagen:** Es necesario observar el histograma de la región de interés que se seleccionó, esto para ver sus propiedades de media, mediana y desviación estándar, con estas propiedades se selecciona el valor de umbral de

Capítulo 3: Implementación de algoritmos para obtención de trayectorias y ajuste de curvas

detección del objeto a seguir, para ver el histograma se hace lo siguiente: Presionar el menú *Window* y a continuación presionar *Histogram*.

- 4) **Ver intensidad de cada color dentro de la región seleccionada:** Para ver la intensidad de cada color RGB de la región de interés, se debe seleccionar el color en el menú *channel* que aparece en el histograma.

Como se mencionó en la sección 2.4, el umbral de tolerancia para la detección de los objetos en el algoritmo de determinación de trayectorias está dado por las ecuaciones (2.2) y (2.3). En la figura 3.1 se puede observar como se aprecia el histograma por colores para la región de interés de una imagen de los videos del motor de vaivén en el programa Photoshop.



Figura 3.1. Histograma de una región de una imagen de los videos de péndulo en Adobe Photoshop. Se observan los tres histogramas de color y sus medidas estadísticas

Herramientas para determinación de regularidad de movimiento articulado mediante el seguimiento en una secuencia de imágenes

La región seleccionada se muestra en la parte superior de la figura, esta región corresponde a los puntos verdes (*targets*) en el brazo, punto medio y en los interruptores de límite establecidos a los lados del mecanismo. En la parte inferior de la figura se pueden observar tres histogramas, uno por cada color RGB, así como los parámetros de estos, los cuales son media, desviación estándar y mediana. Se puede observar que en las regiones de interés seleccionadas domina el color verde, esto se comprueba si se observan los histogramas, en ellos el valor de la media del color verde es mayor que la del rojo y el azul (media rojo = 201.18, media verde = 224.68, media azul = 86.82), lo cual indica que existe mayor intensidad de este color, mientras que el color que tiene menor intensidad en esas regiones es el color azul. En la figura 3.2 se muestran los histogramas para una imagen de los videos de la gimnasta.

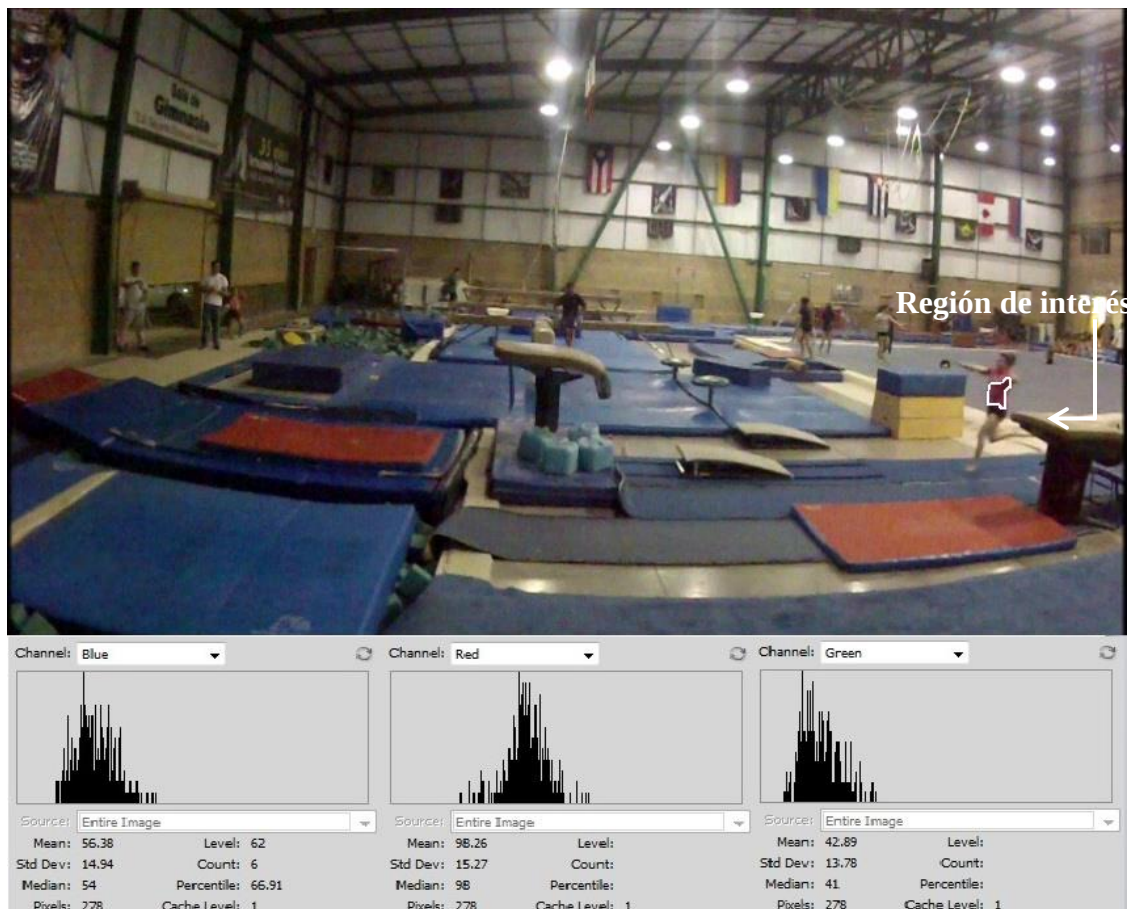


Figura 3.2. Histograma de una región de una imagen de los videos de salto en Adobe Photoshop®.

Capítulo 3: Implementación de algoritmos para obtención de trayectorias y ajuste de curvas

A diferencia de la figura 3.1, en la figura 3.2 se observa que el color que domina en la región seleccionada (uniforme de la gimnasta) es el color rojo, este color presenta un valor de media mayor a la de los colores verde y azul, (media rojo = 98.26, media verde = 42.89, media azul = 56.38). En ambas imágenes el área de interés seleccionada corresponde al objeto que se desea seguir. Utilizando las ecuaciones (2.2) y (2.3), los umbrales T_1 y T_2 para el caso de ambos videos se muestran en la tabla 3.1. Teniendo los valores de umbral de detección de los objetos de interés, se puede proceder a realizar la segmentación para detección de sus centroides en cada imagen. En la siguiente sección se aborda el procedimiento de obtención de trayectorias a partir de una secuencia de imágenes con los umbrales obtenidos mediante el proceso de segmentación.

Tabla 3.1. Valores de media, desviación estándar y umbrales T_1 y T_2 para las regiones de interés de las imágenes correspondientes a los videos de motor tipo péndulo y salto de gimnasta.

Videos/Umbrales por color	Rojo		Verde		Azul	
Saltos de Gimnasta	$\mu = 98.26$		$\mu = 42.89$		$\mu = 56.38$	
	$\sigma = 15.27$		$\sigma = 13.78$		$\sigma = 14.94$	
	$T_1=82.99$	$T_2=113.53$	$T_1=29.11$	$T_2=56.67$	$T_1=41.44$	$T_2=71.32$
Motor tipo péndulo	$\mu = 201.18$		$\mu = 224.68$		$\mu = 86.82$	
	$\sigma = 15.27$		$\sigma = 13.78$		$\sigma = 14.94$	
	$T_1=189.9$	$T_2=216.45$	$T_1=210.9$	$T_2=238.46$	$T_1=71.88$	$T_2=101.76$

3.3 Descripción del proceso de obtención de trayectorias

En esta sección se presenta el método utilizado para llevar a cabo el seguimiento de la trayectoria de objetos, si se desea ver los algoritmos para este procedimiento se puede consultar los apéndices B.1, B.2 y B.3; estos algoritmos se implementaron en los videos de las bases de datos descritas en la sección 2.7 del capítulo anterior. Aquí se describe el procedimiento para dos conjuntos de videos, en primer lugar para el caso del motor de vaivén (ver figura 3.3.(a)) y en segundo lugar en el caso de los videos de la gimnasta al realizar un salto de entrenamiento (ver figura 3.5). La prueba de la gimnasta tiene un mayor grado de complejidad puesto que se trata de un “objeto” de región variable, los movimientos de una persona pueden cambiar la forma en que esta es percibida por la cámara; además estos videos están en un ambiente no controlado y con variaciones de iluminación en las diferentes zonas del mismo. A continuación se presenta el pseudocódigo del algoritmo del apéndice B.1, los siguientes párrafos describen detalladamente el proceso.

PSEUDOCÓDIGO PARA OBTENCIÓN DE TRAYECTORIAS

Función obtenerCentroides(imagen de entrada)

n = número total de imágenes del video

Para i = 1 hasta n
{

Abrir imagen:

Imagen = leer imagen de entrada;

Descomposición de imagen RGB en sus tres componentes:

Componente rojo R = Imagen(componente rojo)

Componente verde G = Imagen(componente verde)

Componente azul B = Imagen(componente azul)

Umbralado y binarización:

N = ancho de la imagen en turno

M = Alto de la imagen en turno

Capítulo 3: Implementación de algoritmos para obtención de trayectorias y ajuste de curvas

```
Y = matriz de ceros de tamaño N*M
%%Y será una imagen totalmente negra usada como fondo%%

T1R = umbral inferior Rojo
T2R = umbral superior Rojo
T1G = umbral inferior Verde
T2G = umbral superior Verde
T1B = umbral inferior Azul
T2B = umbral superior Azul

Repetir desde i=1 hasta N y desde j=1 hasta M para revisar
cada pixel de la imagen

Se evalúa pixel (i,j)
Si((T1R > componente rojo de pixel (i,j) < T2R) and (T1G >
componente verde de pixel (i,j) < T2G) and (T1B > componente
azul de pixel (i,j) < T2B))

Entonces Y(i,j) = 1;

De lo contrario Y(i,j) = 0;

Se aplican operaciones morfológicas,
Cerradura = Dilatación + Erosión:
Y = imclose(Y);

Si un grupo de pixeles es menor que un número establecido GP
se eliminan pixeles dispersos:
Y = bwareaopen(Y,GP);

Etiquetado de objetos
bwlable(Y);

Generación de tabla de centroides y tiempo (X,Y,t)
Centroides = regionprops(Y,'centroides');

Si se encuentra al menos un centroide en imagen en turno
Entonces asigna índice "i" a imagen en turno.

Pasa a la siguiente imagen para analizar
}
```

En la **tabla 3.2** se muestran las funciones de Matlab utilizadas en el algoritmo edp_Al y su descripción.

Herramientas para determinación de regularidad de movimiento articulado mediante el seguimiento en una secuencia de imágenes

Tabla 3.2. Funciones de Matlab utilizadas en algoritmo “edp_AI” y su descripción.

Función	Descripción
<i>imclose(imagen)</i>	Aplica cerradura, consistente en una operación de dilatación, seguida por una operación de erosión, se utiliza para rellenar y definir objetos de interés.
<i>bwareaopen(imagen, GP)</i>	En una imagen binaria, elimina grupos de pixeles conectados cuyo número sea menor a un límite GP establecido, a esta operación se le conoce como apertura de área.
<i>bwlabel(imagen)</i>	Retorna una matriz que contiene etiquetas de los objetos encontrados en una imagen binaria.
<i>regionprops(imagen, atributos)</i>	Para una imagen binaria, regresa los atributos establecidos en la función, entre estos atributos se encuentran: Área, centroide, <i>bounding box</i> , etc.

En la figura 3.3, se muestra una descripción ilustrativa del proceso de obtención de centroides para una imagen del mecanismo de motor con movimiento de vaivén.

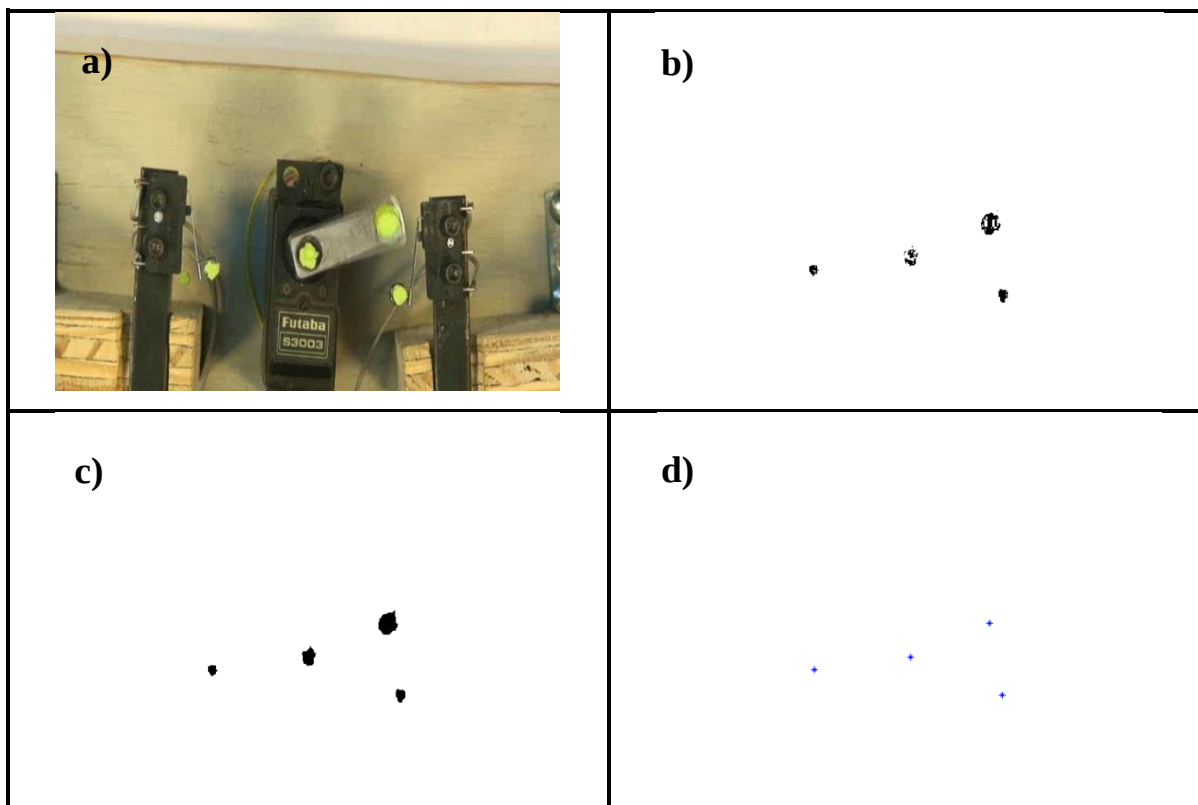


Figura 3.3. a) Motor de CD con movimiento de vaivén, b) Imagen binaria obtenida de la segmentación, c) Imagen binaria con regiones de interés rellenas, d) Centroides graficados de las regiones de interés.

Capítulo 3: Implementación de algoritmos para obtención de trayectorias y ajuste de curvas

Durante todo el proceso se trabajó con imágenes a color RGB como la de la figura 3.3(a). Estas imágenes se separaron en las tres matrices, una por color. A partir de los umbrales obtenidos T_1 y T_2 (ver sección 3.2) se llevó a cabo el proceso de segmentación, de la siguiente manera: Si la intensidad de un pixel determinado está dentro los valores de umbral T_1 y T_2 para los tres colores que componen la imagen, ese pixel tendrá un valor de 1 (blanco), mientras que si la intensidad esta fuera de esos valores, se le asigna un valor de 0 (negro); el resultado es una imagen binaria (Y) como la de la figura 3.3(b). Se puede observar que hay pixeles dentro de las regiones de interés que no caen dentro de los límites de umbral establecidos, pero aun así se sabe que son parte de los objetos seguidos, por lo que deben ser rellenadas. Para esto se utilizan operaciones morfológicas de dilatación, erosión y cerradura, Matlab incluye la función **imclose** para la aplicación de estas operaciones (Ver apéndice A.1 para una definición de estos conceptos). Con esto se obtiene la imagen de la figura 3.3(c), en la cual se pueden observar las regiones de interés de una manera más definida. Posteriormente, se aplica la función **regionprops** para detectar los centroides de las regiones de interés que quedan, esta función retorna una tabla con las coordenadas de los centroides detectados. Para finalizar el proceso, se aplica la función **bwlabel**, esta función retorna un conjunto de etiquetas correspondientes a cada objeto detectado, de manera que se tenga una referencia del objeto de una imagen con respecto a la siguiente, **bwlabel** también retorna el número de objetos detectados. La imagen resultante con los centroides graficados se muestra en la figura 3.3(d); esta figura presenta los centroides de los puntos de interés para una sola imagen, sin embargo, para graficar la trayectoria de dichos puntos se necesitan los centroides para toda la secuencia de imágenes del video, de modo que el mismo procedimiento se aplica a todos los cuadros del video, resultando la gráfica mostrada en la figura 3.4. Otro dato muy importante que resulta de este proceso es el tiempo (t) en el cual se obtiene cada centroide, este tiempo es relativo a la primera imagen del video y se obtiene gracias a la captura del índice de cuadro **i** de cada imagen procesada, conociendo este índice y la velocidad de cuadro del video (Ver tabla 2.1 en la sección 2.72.8), se puede conocer el tiempo en el cual cada centroide es capturado mediante la ecuación (3.1).

$$t = \frac{i}{\text{velocidad de cuadro}} \quad (3.1)$$

Herramientas para determinación de regularidad de movimiento articulado mediante el seguimiento en una secuencia de imágenes

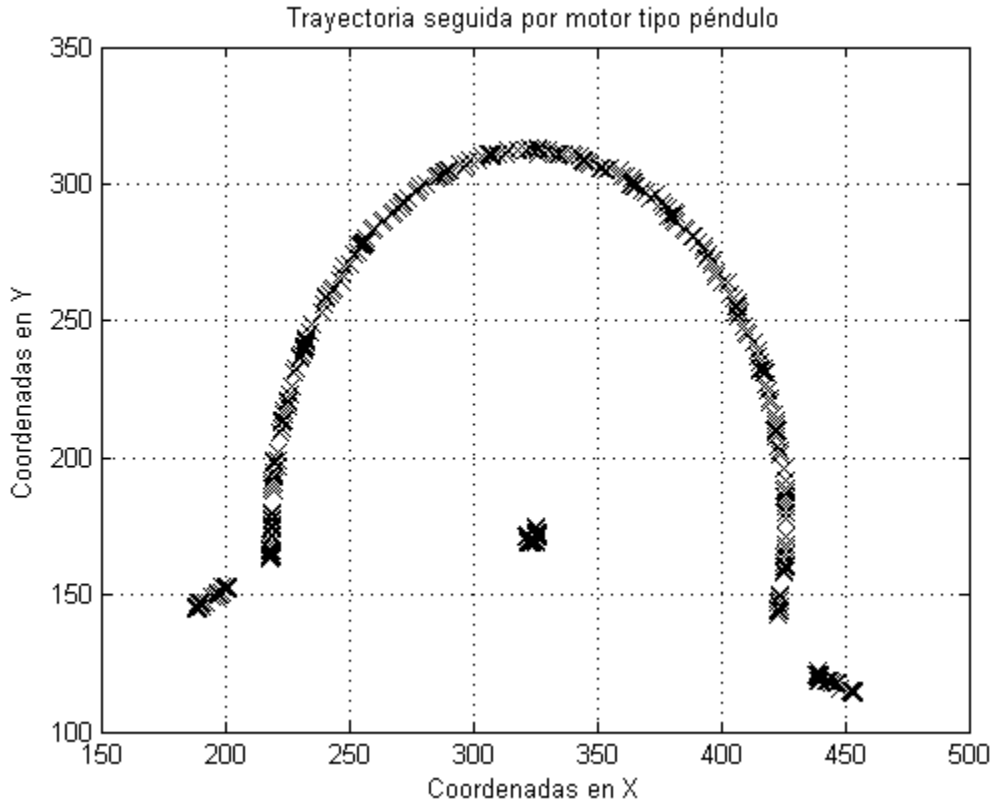


Figura 3.4. Grafica de todos los puntos centroides que deben corresponder a la trayectoria seguida por un grupo de objetos asignados a un motor con movimiento de vaivén.

Los procedimientos mencionados en este capítulo se aplicaron a todos los videos utilizados como base de datos (ver sección 2.7) para ser analizados. En el caso de los videos de salto de la gimnasta, las condiciones de umbral fueron las que corresponden al uniforme de la gimnasta. El resultado obtenido al graficar la trayectoria y sobreponerla con una de las fotos se muestra en la figura 3.5. Los puntos que marcan la trayectoria pueden aislarse del fondo de la imagen que no es de interés para efectos de análisis, además, aislando la trayectoria seguida, se puede reducir en gran medida el peso de los datos, puesto que en vez de tener un video completo, solo se tendrá una tabla con los datos (X,Y,t) de los puntos obtenidos, donde t es el tiempo de cada centroide dado por la ecuación (3.1); la trayectoria aislada para un salto de la gimnasta se muestra en la figura 3.6. En el apéndice C se muestran las tablas de datos (X,Y,t) para los videos de motor y saltos.



Figura 3.5. Gráfica de la trayectoria seguida por una gimnasta al realizar un salto.
Trayectoria con sub-muestreo de 100 imágenes.

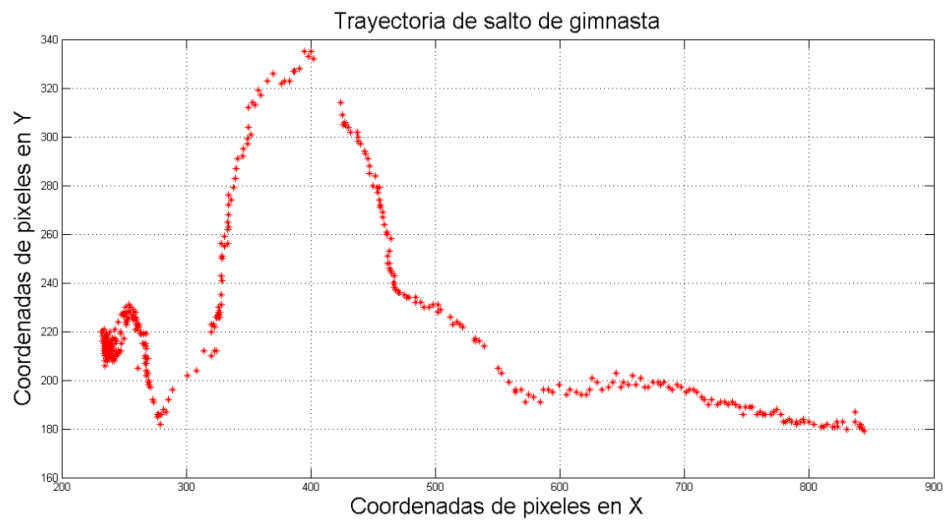


Figura 3.6. Gráfica aislada de la trayectoria seguida por una gimnasta al realizar un salto.
Trayectoria sin sub-muestreo.

Herramientas para determinación de regularidad de movimiento articulado mediante el seguimiento en una secuencia de imágenes

Las trayectorias obtenidas correspondientes a los tres videos tomados se nombraron salto A, B y C. Los saltos A y C son del mismo tipo, mientras que en B se realizó un tipo de salto distinto, en la figura 3.7 se pueden observar las tres trayectorias graficadas. Cada punto mostrado en la figura 3.7 tiene una coordenada (X, Y, t) que indica la posición del pixel correspondiente a cada imagen del video así como el tiempo relativo al inicio del video en el cual la gimnasta pasó por ese punto (Ver apéndice C). Los tres videos de saltos tienen una duración diferente, pero se recortaron al tiempo suficiente para captar el salto desde su inicio hasta su fin. Los saltos no son idénticos, pues además de tener trayectorias distintas, algunos se ejecutan con mayor o menor rapidez, todo esto ocasiona que cada video tenga una cantidad distinta de imágenes y por lo tanto una cantidad de puntos diferente entre el inicio y el final del salto. En las tres trayectorias graficadas en la figura 3.7 se toma como tiempo cero al tiempo del cuadro en el cual la gimnasta aparece en el video.

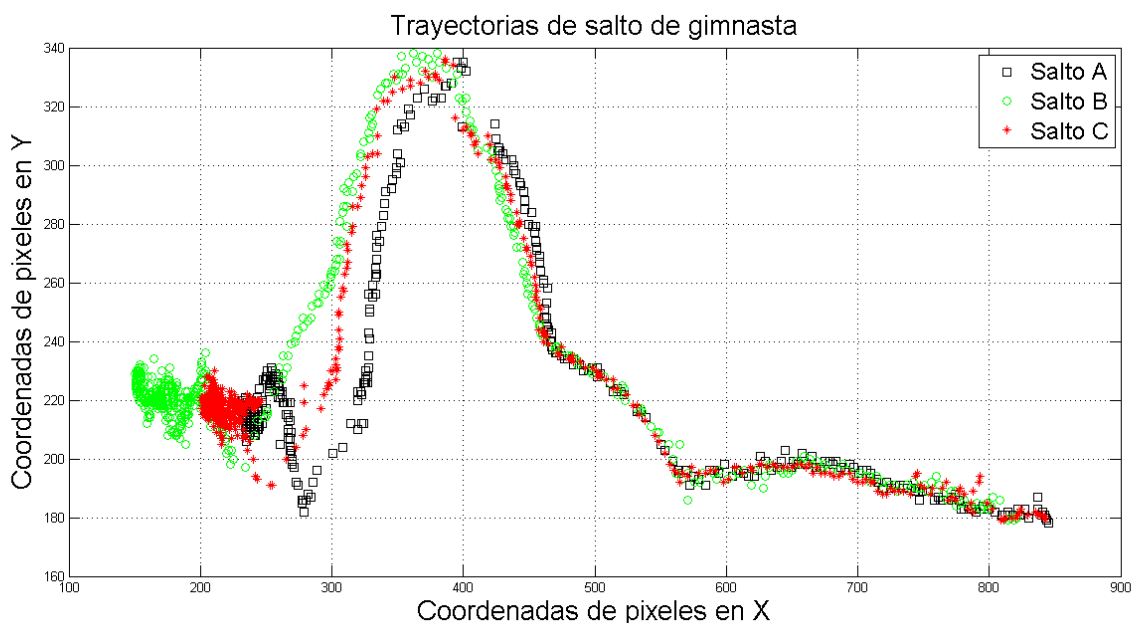


Figura 3.7. Gráfica de las trayectorias seguidas por la gimnasta en tres saltos.

3.4 Problemas encontrados durante el seguimiento y la generación de trayectorias

Al ejecutar el algoritmo de detección de objetos y generación de trayectorias, se presentaron algunos problemas. Por ejemplo, en algunas de las imágenes del video no se detecta la presencia de ningún objeto de interés incluso cuando el objeto sí está presente (falsos negativos), esto es debido a factores como obstrucciones del objeto, cambios de iluminación en una zona determinada y resolución pobre del video, entre otros. También se presenta el caso en el que en algunas imágenes del video ocurren detecciones en zonas donde no está presente el objeto de interés a seguir (falsos positivos), éste problema se debe a que existen regiones de una determinada imagen, en donde los valores de intensidad de los pixeles caen dentro del rango de valores de umbral establecidos.

A continuación se presentan tres tablas correspondientes a los tres conjuntos de videos usados como bases de datos. En estas tablas se presenta el porcentaje de errores debidos a falsos negativos que se tuvieron en cada uno de los videos, considerando el total de imágenes de estos. La tabla 3.3 se enfoca en los videos de la gimnasta. La tabla 3.4 muestra el análisis para tres videos representativos del conjunto de diez videos que se tomaron al motor con movimiento de tipo péndulo, el nombre del video corresponde a la configuración de lámparas encendidas y apagadas de la mesa en la cual se ubicó el mecanismo del motor para hacer las tomas de los videos; en el Capítulo 4 se explica más a detalle el proceso llevado a cabo para la toma de los videos de motor tipo péndulo y el análisis realizado de estos videos, mientras tanto se puede observar la figura 4.1 si se quiere tener una mejor idea de cómo están ubicadas las lámparas. En la tabla 3.5 se considera el video del robot Fanuc. La cantidad de cuadros de imagen depende de la duración de cada video conteniendo al objeto de interés

Herramientas para determinación de regularidad de movimiento articulado mediante el seguimiento en una secuencia de imágenes

Tabla 3.3. Porcentajes de falsos negativos para videos de gimnasta, considerando la cantidad total de imágenes por video.

Videos de gimnasta	Cantidad de imágenes	Falsos negativos
Salto A	499	5.4%
Salto B	633	10.1%
Salto C	774	6.8%

Tabla 3.4. Porcentajes de falsos negativos para videos de péndulo, considerando la cantidad total de imágenes por video.

Videos de péndulo: Configuración de lámparas	Cantidad de imágenes	Falsos negativos
Todas encendidas	309	0.64%
Lámparas Izquierda apagadas	310	0.24%
Lámparas derecha apagadas	323	0%

Tabla 3.5. Porcentajes de falsos negativos para video de robot Fanuc, considerando la cantidad total de imágenes del video.

Videos de péndulo	Cantidad de imágenes	Falsos negativos
Video robot Fanuc	710	4.54%

Capítulo 3: Implementación de algoritmos para obtención de trayectorias y ajuste de curvas

En la mayoría de los videos se presentan errores debidos a falsos negativos, pero los videos en los cuales se tiene un mayor porcentaje de este problema son los de la gimnasta, debido a que en estos videos el ambiente no es controlado, la iluminación varía de una zona a otra de las imágenes, además de que no se tienen *targets* de detección, simplemente se utiliza el color del traje deportivo de la gimnasta como rasgo a seguir. Por su parte, en los videos de péndulo y robot Fanuc, además de contar con targets seleccionados de un color contrastante, se tiene un ambiente más controlado, con una iluminación mayormente constante para todas las zonas del video, es por ello que se tienen porcentajes menores de falsos negativos, incluso en uno de los videos del péndulo no se obtuvo ningún falso negativo. La figura 3.8 muestra algunas de las imágenes en las cuales se presenta el problema de falsos negativos para los videos de la gimnasta, algunos de ellos ocasionados por obstrucciones del color a seguir (la gimnasta obstruye su traje con el brazo), mientras que otros son ocasionados por las variaciones en iluminación del video (se tienen sombras en la región de interés). Estas imágenes fueron extraídas del video de salto A, el cual tiene una cantidad de 499 imágenes en total. En la figura 3.9 se ilustra el problema de falsos positivos, es decir, puntos detectados en zonas donde no se encuentra el objeto de interés a seguir.

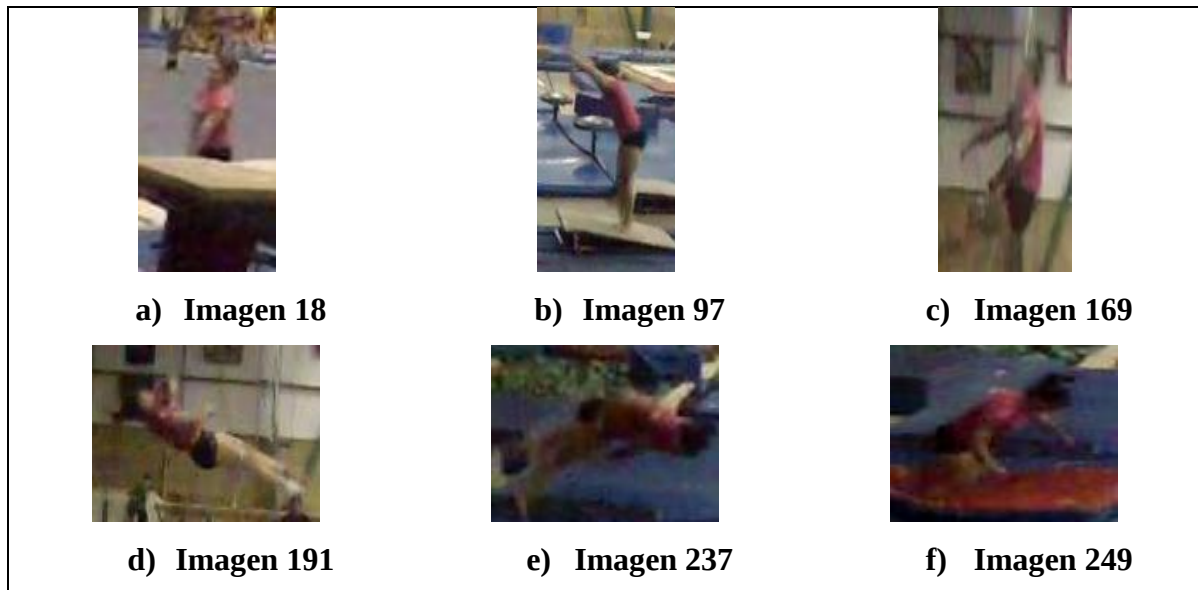


Figura 3.8. Ejemplos de imágenes con detección fallida (falsos negativos).

Herramientas para determinación de regularidad de movimiento articulado mediante el seguimiento en una secuencia de imágenes

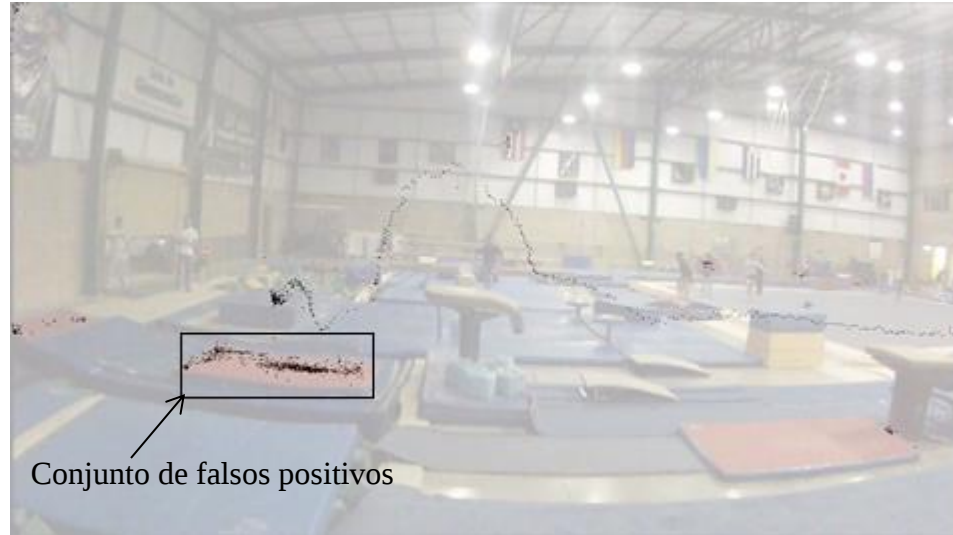


Figura 3.9. Muestra de trayectoria con falsos positivos.

En la figura 3.9 se observa como existe una cantidad considerable de puntos en zonas por donde la gimnasta nunca pasa, como por ejemplo el colchón donde cae (recuadro señalado), esto se debe a que estas zonas caen dentro del umbral de color elegido para detectar el uniforme de la gimnasta. Para eliminar la mayor cantidad posible falsos positivos, cuidando a la vez tener el menor porcentaje de falsos negativos posible, se tomaron las siguientes medidas: En primer lugar, se intensificó la sintonización del parámetro de umbral de detección de las regiones de interés, esto se hizo obteniendo los umbrales T_1 y T_2 utilizando valores promediados de las propiedades estadísticas de los histogramas de varias imágenes del video, esto tratando de utilizar imágenes de todo el video (inicio, mitad y final), en todos los casos el histograma dio un rango de umbrales más amplio al necesitado, lo cual ocasionó falsos positivos, durante la sintonización se fue acortando este rango, aumentando poco a poco el valor de T_1 y disminuyendo el valor de T_2 hasta encontrar un rango que permitiera la menor cantidad de falsos positivos posible, sin ocasionar el aumento de falsos negativos. Se intensificó también la sintonización de la operación de apertura de área para eliminar las áreas que no pertenecieran a las regiones de interés, cuidando también de no ocasionar falsos negativos. Se asignaron restricciones de detección por zonas de la imagen, analizando solamente aquellas zonas en las cuales se

sabe de antemano que el o los objetos de interés tendrán interacción y dejando fuera del análisis las zonas restantes.

Aún con la aplicación de estos dos métodos, se siguieron presentando problemas de falsos positivos, puesto que no es sencillo sintonizar perfectamente los parámetros de umbral y operaciones morfológicas para que funcionen en todas las imágenes de un video, en especial si el video se tomó en un ambiente no controlado, también es complicado establecer restricciones de zona cuando los falsos positivos se presentan en zonas cercanas a las regiones de interés. Debido a estos problemas, se implementó un tercer método pero interactivo, el cual resultó ser eficaz. Este método consiste en que, una vez teniendo todos los puntos detectados (incluyendo falsos positivos), se deben seleccionar aquellos puntos que son útiles y desechar los que no lo son mediante un algoritmo de selección manual. El algoritmo, llamado **selectPoints_AI** (Ver Apéndice B.3 para observar el código diseñado), imprime todos los puntos de la trayectoria en una gráfica incluyendo los falsos positivos, como se aprecia en la figura 3.10(a) una vez teniendo esta gráfica, se tiene la opción de seleccionar manualmente aquellos puntos que pertenecen a la trayectoria, dejando fuera los falsos positivos, el resultado se aprecia en la figura 3.10(b).

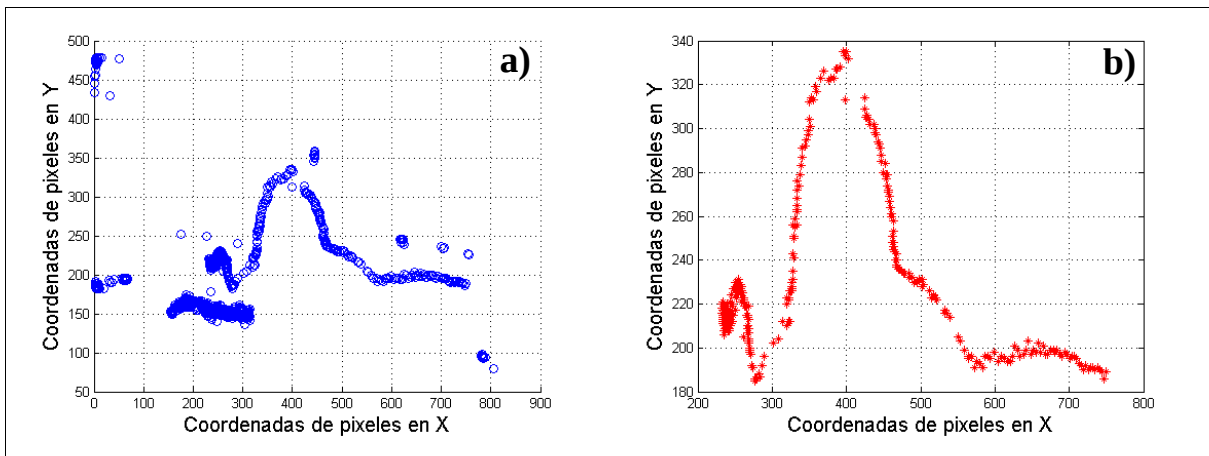


Figura 3.10. Ilustración del algoritmo interactivo **selectPoints_AI** para eliminación de falsos positivos. a) Trayectoria antes de aplicar algoritmo, b) trayectoria después de aplicar algoritmo.

Herramientas para determinación de regularidad de movimiento articulado mediante el seguimiento en una secuencia de imágenes

Tanto para el caso del motor como para el del robot Fanuc, no se mostró ningún falso positivo, lo que indica de nuevo que los algoritmos funcionan de mejor manera en ambientes controlados. En el siguiente capítulo se muestran dos formas de caracterizar las trayectorias obtenidas, de manera que se pueda obtener una función matemática que describa estas trayectorias para poder hacer comparaciones entre sus parámetros.

3.5 Comparación de técnicas usadas para el ajuste de trayectorias

En esta sección se plantea una comparación entre las técnicas de red neuronal e interpolación polinomial utilizadas en este proyecto de investigación para ajuste de curvas de trayectoria. Para iniciar el capítulo, se aborda una técnica basada en una red neuronal, por lo que la primera mitad del mismo se enfoca en describir como se llevó a cabo la selección de los parámetros necesarios para el ajuste de las trayectorias mediante este algoritmo. Por su parte, la segunda mitad del capítulo examina una técnica de interpolación para lograr el mismo resultado que la red neuronal, esta técnica es más simple que la primera matemáticamente hablando, pero tiene la bondad de necesitar un menor costo computacional. Para finalizar el capítulo, se evalúan los resultados obtenidos al aplicar las dos técnicas mencionadas, para tomar una decisión de cuál de ellas conviene más que sea utilizada en actividades posteriores.

3.5.1. Implementación de una red neuronal para ajuste de curvas de trayectorias

En la figura 3.4 y figura 3.6 se mostró la trayectoria recorrida por un motor con movimiento de vaivén y la trayectoria seguida por una gimnasta al realizar un salto respectivamente, mismas que se tomarán para la implementación de la red. Para explicar el proceso de selección de parámetros se utiliza la trayectoria del motor de vaivén, en este caso el video se sometió a sub-muestreo de 50 imágenes para agilizar la ejecución de los algoritmos. En figura 3.11 se muestra la trayectoria del motor de vaivén para un sub-

Capítulo 3: Implementación de algoritmos para obtención de trayectorias y ajuste de curvas

muestreo de 50 imágenes. Los datos de entrada de la red son los valores de las coordenadas horizontales de los puntos de la figura (X), mientras que los datos de salida son los valores de las coordenadas verticales de los puntos (Y). La red trata de predecir el modelo de la trayectoria, de manera que con los puntos (X,Y) dados, se pueda conocer cualquier otro punto de la curva. El apéndice B.4 muestra el algoritmo utilizado para la selección de parámetros y entrenamiento de la red neuronal utilizada en el presente trabajo.

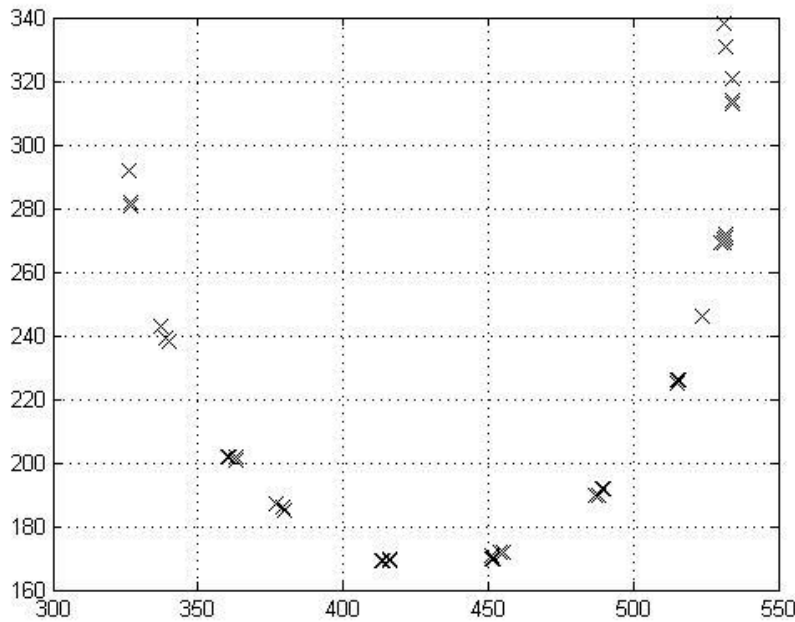


Figura 3.11. Trayectoria del motor de vaivén graficada en coordenadas (X, Y). Trayectoria obtenida utilizando un sub-muestreo de 50 imágenes.

Se puede observar que la trayectoria de la figura 3.11 es diferente en apariencia a la de la figura 3.4, esto es debido a que la figura 3.11 está graficada en las coordenadas (X,Y) convencionales, mientras que en la figura 3.4 las coordenadas graficadas corresponden a las de una imagen, y las coordenadas de una imagen son diferentes al sistema de coordenadas convencional (ver sección 2.2). Sin embargo, esta diferencia entre las imágenes no afecta de ninguna manera los resultados, lo único que cambia es la apreciación de las gráficas.

Existen tres parámetros fundamentales para el buen funcionamiento de la red neuronal aquí presentada, el primero es el número de capas internas de la red así como el número de unidades por capa, en segundo lugar está la tasa de aprendizaje de la red (η), y

Herramientas para determinación de regularidad de movimiento articulado mediante el seguimiento en una secuencia de imágenes

en tercer lugar está la estrategia de aprendizaje a utilizar. Para seleccionar los parámetros se ejecutó el algoritmo del apéndice B.4 una cierta cantidad de veces y se registró precisión obtenida para combinaciones de valores de estos parámetros mediante el uso de un ciclo *for*. Todos los parámetros de la red se establecen mediante la función *net1 = newff(parámetros)*, esta función te permite crear una red neuronal, la cual se nombró “net1” y establecerle los parámetros de sintonización. El 70% de los centroides se tomaron como datos de entrenamiento y el 15% se tomaron como datos validación y el otro 15% como datos de prueba para medir la precisión de la red, para estos se utilizaron los métodos:

net1.divideparam.trainRatio = 75;

net1.divideparam.valRatio = 15;

net1.divideparam.testRatio = 15;

La red neuronal se entrena mediante la función *train(net1)*. El primer parámetro a sintonizar fue el número de capas internas de la red y el número de unidades por capa, entre mayor sean estos dos números, más compleja es la red internamente. Al sintonizar este parámetro resultó una red de una sola capa interna con 20 unidades para el caso de la trayectoria del motor. La tasa de aprendizaje (*n*) es la tasa a la cual la red neuronal aprende los pesos *w*, estos pesos conforman el polinomio en el cual se substituye la entrada *X* para obtener un resultado *Y*. La tasa de aprendizaje determina en gran medida el factor de rapidez vs precisión de entrenamiento de la red; para seleccionar este parámetro, el algoritmo del apéndice B.4 se ejecutó 10 veces cada uno de los valores de *n* entre [0.1,1] con incrementos de 0.1, y se calculó un promedio de las precisiones obtenidas, seleccionando la tasa de aprendizaje cuyo promedio de precisión fue mayor. (Véase tabla 3.6).

Las redes neuronales pueden ser sintonizadas de manera que funcionen con distintas estrategias de aprendizaje, las estrategias de aprendizaje son una serie de ecuaciones que sirven como funciones internas de cada neurona de la red, en el presente trabajo se evaluó el entrenamiento de la red neuronal para cinco estrategias de aprendizaje distintas, de nuevo se obtuvo un promedio a partir de 10 ejecuciones para cada estrategia y para cada tasa de aprendizaje *n*, para lograr todas las combinaciones de estrategias de aprendizaje y tasas de

Capítulo 3: Implementación de algoritmos para obtención de trayectorias y ajuste de curvas

aprendizaje con rango de [0.1,10], el algoritmo de entrenamiento de la red se tuvo que ejecutar un total de 500 veces, a continuación se muestran las 5 estrategias de aprendizaje utilizadas y en la tabla 3.6 se muestran las precisiones obtenidas:

- *Levenberg-Marquardt backpropagation (trainlm).*
- *Quasi-Newton Backpropagation (trainbfg).*
- *Porlak Ribiere Conjugate Gradient (trainpr).*
- *Gradient Descent Backpropagation (traingd).*
- *Scale Conjugate gradient backpropagation (trainscg).*

Tabla 3.6. Precisiones de ajuste de trayectoria para distintos valores de la tasa de aprendizaje n y diferentes estrategias de aprendizaje.

Ejecución	Valor de n	trainlm	trainbfg	trainrp	traingd	trainscg	Promedio
1	0.1	99.20%	95.17%	71.31%	31.22%	91.52%	90.28%
2	0.2	95.52%	92.91%	90.76%	53.37%	89.72%	84.46%
3	0.3	86.34%	89.30%	91.17%	62.90%	89.04%	83.75%
4	0.4	97.30%	92.82%	88.12%	16.10%	90.46%	76.96%
5	0.5	96.32%	82.53%	87.65%	40.38%	82.86%	77.95%
6	0.6	98.09%	88.37%	79.68%	57.15%	80.00%	80.66%
7	0.7	87.89%	62.59%	90.18%	29.34%	90.79%	72.16%
8	0.8	91.88%	81.19%	57.30%	55.11%	91.08%	75.31%
9	0.9	98.98%	83.26%	90.34%	61.28%	97.53%	86.28%
10	1	98.32%	93.82%	89.67%	54.31%	88.34%	84.89%
Promedio		94.98%	86.20%	83.62%	46.12%	89.13%	

Herramientas para determinación de regularidad de movimiento articulado mediante el seguimiento en una secuencia de imágenes

Como puede observarse, los promedios más altos de precisión en la regresión están dados para la estrategia de *Levenberg-Marquardt backpropagation (trainlm)* y para la tasa de aprendizaje $n=0.1$, estos fueron los parámetros elegidos para entrenar la red neuronal, por lo que se tuvo una precisión de 94.98%. En la figura 3.12 se muestra la comparación entre la trayectoria ajustada y los puntos que se tenían en la figura 3.11.

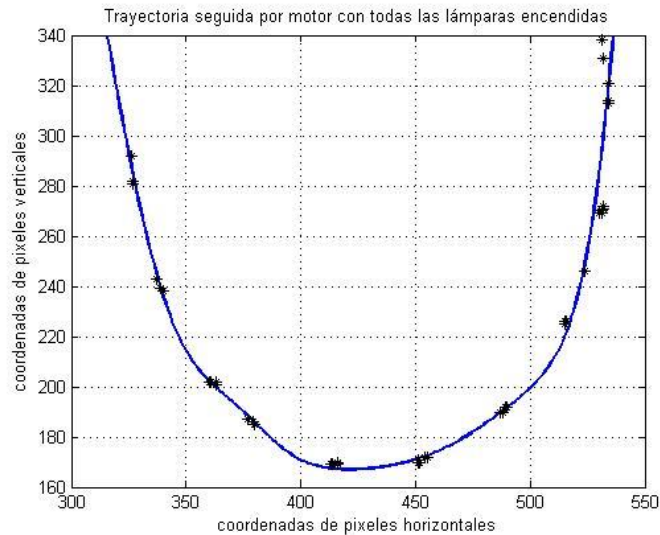


Figura 3.12. Ajuste de trayectoria de motor con movimiento de vaivén y su comparación con los puntos de referencia.

En la figura 3.13 se muestra el resultado del método de red neuronal pero aplicado a uno de los saltos de la gimnasta, de nuevo se observa que la trayectoria está “de cabeza”, pero de nuevo esto es simplemente por efecto del formato de coordenadas convencionales y coordenadas de imagen, para corregir esto se puede restar al tamaño vertical de la imagen el valor de cada punto en Y (en este caso se tienen imágenes de 840x480 píxeles) de la siguiente manera:

$$Y = W - Y$$

Donde Y es el conjunto de coordenadas verticales de los centroides del objeto seguido y W es el tamaño vertical de la imagen (en este caso 480 píxeles). La máxima precisión obtenida en los videos de saltos fue de 98.00% para el caso del salto A.

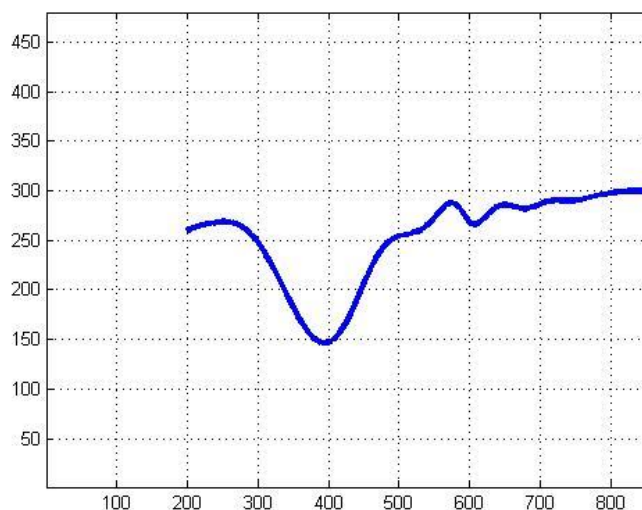


Figura 3.13. Ajuste de trayectoria seguida por gimnasta al realizar un salto mediante red neuronal.

En la tabla 3.7 se muestran los pesos w para las redes neuronales obtenidas de las trayectorias del motor y la gimnasta, con estos pesos se pueden generar las trayectorias de la figura 3.12 y la figura 3.13.

Tabla 3.7. Conjunto de pesos para las redes neuronales entrenadas para uno de los videos de motor y de salto de gimnasta.

Video	Pesos
Motor	[-1.8459 -3.7674 -0.5919 -0.4254 0.3333 3.6452 -1.4537 0.7009 -1.1186 0.2809]
Gimnasta	[-0.2047 -0.0915 -0.8126 0.9753 -0.5361 0.0199 -0.3882 - 0.7946 0.0458 -0.4227 0.1254 0.4342 0.3498 -0.8061 -2.9683 1.4210 -0.3165 1.1633 1.8519 -0.4157]

Herramientas para determinación de regularidad de movimiento articulado mediante el seguimiento en una secuencia de imágenes

3.5.2. Implementación de algoritmo de mínimos cuadrados para ajuste de curvas de trayectorias

Como segunda técnica utilizada para ajuste de curvas, se implementó un algoritmo consistente en el método de mínimos cuadrados para crear un polinomio que caracterizara las trayectorias. Al inicio de esta técnica, los puntos de las trayectorias deben ser pre-procesados mediante normalización, para esto se obtiene la media y la desviación estándar de todo el conjunto de puntos, teniendo estos valores, se aplica la ecuación (3. 1).

$$X_{Norm} = \frac{X - \mu}{\sigma} \quad (3. 1)$$

Donde X es el conjunto de datos a normalizar, μ es la media de estos datos y σ su desviación estándar, mientras que X_{Norm} son los datos normalizados. Los únicos datos que deben ser normalizados son los datos en X , mientras que los datos en Y son tratados como están originalmente. Una vez normalizados, los datos son sometidos al algoritmo de ajuste mediante una función de Matlab llamada `polyfit`, esta función implementa la técnica de ajuste por medio del método de mínimos cuadrados, (ver sección 2.5.2), y su sintaxis es la siguiente:

```
[p] = polyfit(X_Norm, Y, orden);
```

En esta línea de código, p es el vector de coeficientes del polinomio obtenido, X_{Norm} son los datos en X normalizados, Y son los datos para los cuales se lleva a cabo el ajuste, y “orden” indica el orden del polinomio p . Si se evalúa el polinomio p , se puede obtener la diferencia entre los valores reales de Y y los valores obtenidos como resultado de la evaluación para saber qué tan bien el polinomio p ajusta los puntos de la trayectoria, a partir de estas diferencias se puede obtener el error que existe entre la trayectoria real y la interpolada. La evaluación se ejecuta mediante la siguiente línea de código:

```
YY = polyval(p, X_Norm);
```

Donde YY es el resultado de la evaluación, p es el polinomio obtenido de la función `polyfit` y X_{Norm} son los datos en X normalizados. Al igual que para el caso de la red neuronal, en

Capítulo 3: Implementación de algoritmos para obtención de trayectorias y ajuste de curvas

este caso el error utilizado para evaluar la eficiencia del ajuste es el ECM. Para obtener el ECM se calcula la ecuación (2.5) vista en la sección 2.5.3. En la figura 3.14 se observa la trayectoria ajustada para un conjunto de puntos correspondientes al salto A. Comparando la trayectoria de la figura 3.13 con la trayectoria de la figura 3.14, se puede observar que son muy similares, aunque no idénticas, puesto que cada algoritmo tiene su propio método y sus propios errores.

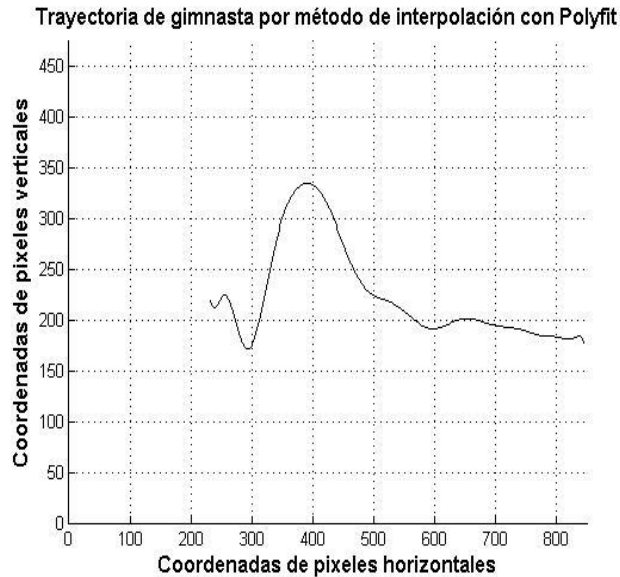


Figura 3.14. Ajuste mediante mínimos cuadrados de trayectoria seguida por gimnasta al realizar un salto.

Las ecuaciones polinomiales para la trayectoria correspondiente al video del motor de vaivén con todas las lámparas encendidas y la correspondiente al video de salto A de la gimnasta se muestran en la tabla 3.8. Se puede observar que la ecuación correspondiente al salto de la gimnasta es de un grado significativamente más alto que la ecuación de la trayectoria del motor. Para elegir el orden de los polinomios, se creó un ciclo y se ejecutó el algoritmo del apéndice B.5 n veces aumentando el orden, para cada ejecución se obtuvo un polinomio y con este polinomio se obtuvo un error cuadrático medio, para cada aumento del valor de orden el ECM fue disminuyendo y se tenía un polinomio más complejo. El orden elegido fue aquel en el cual el ECM no disminuyó significativamente al aumentar el orden un valor más, se tomó como valor poco significativo una disminución del valor de error cuadrático medio menor o igual a 10.

Herramientas para determinación de regularidad de movimiento articulado mediante el seguimiento en una secuencia de imágenes

Tabla 3.8. Ecuaciones polinomiales para una de las trayectorias seguidas por el motor y una trayectoria seguida por la gimnasta al hacer un salto.

Tipo de video	Ecuación polonomial
Motor	$Y = 5.99x^8 - 6.9x^7 - 56.37x^6 + 6.31x^5 + 71.19x^4 + 0.4x^3 - 69.92x^2 - 2.33x + 313.85$
Gimnasta	$Y = 22.72x^{19} - 401.14x^{18} + 3003x^{17} - 12140x^{16} + 26967x^{15} - 24468x^{14} - 25013x^{13} + 84800x^{12} - 55771x^{11} - 52381x^{10} + 86143x^9 - 8991.9x^8 - 39871x^7 + 14898x^6 + 5768.3x^5 - 2156.6x^4 + 63.64x^3 - 643.79x^2 + 33.48x + 333.79$

Todo lo necesario para implementar el método de mínimos cuadrados se implementó en una función la cual se nombró **interpolación_A1** y se muestra en el apéndice B.6.

3.5.3. Comparación de resultados de algoritmos de ajuste de curvas de trayectorias

Una vez aplicadas las dos técnicas mencionadas en este capítulo, se llevó a cabo una evaluación para ver cuál de ellas sería implementada en análisis posteriores, esta evaluación se dio mediante la obtención del error cuadrático medio de los dos algoritmos al implementarlos en los tres videos de la gimnasta, los resultados se muestran en la tabla 3.9. Para hacer las comparaciones se utilizó la misma cantidad de puntos en las trayectorias analizadas.

Tabla 3.9. Comparación del error cuadrático medio (ECM) de los métodos de red neuronal e interpolación.

Salto de gimnasta	ECM Red neuronal	ECM Mínimos Cuadrados
A	11.72	25.21
B	502.39	20.82
C	316.46	17.80

En la tabla 3.9 se puede observar que si bien la red neuronal tuvo un menor error cuadrático medio en el caso del salto A, no fue así para los casos de saltos B y C, de hecho se tiene una gran diferencia de error en estos dos saltos. En el caso de la técnica de mínimos cuadrados, los valores de error son más cercanos unos con respecto a otros, indicando un algoritmo más consistente que obtiene resultados de rangos similares para diferentes trayectorias. Otra ventaja que se observó de la técnica de mínimos cuadrados con respecto a la red neuronal, es que la primera puede ser ejecutada fácilmente en un solo paso, mientras que la red necesita de dos pasos para ser implementada, los cuales son entrenamiento y predicción. Además de la evaluación utilizando el ECM, los algoritmos también se compararon en cuanto a tiempo de ejecución, Para medir el tiempo de ejecución ambos algoritmos se ejecutaron bajo las mismas condiciones (mismos datos y variables), y se activó la herramienta de Matlab “*Run and time*” que el tiempo de ejecución de los algoritmos, los resultados arrojaron un menor tiempo de ejecución para el caso del algoritmo de interpolación como se muestra en la tabla 3.10.

Herramientas para determinación de regularidad de movimiento articulado mediante el seguimiento en una secuencia de imágenes

Tabla 3.10. Tiempos de ejecución de algoritmos en base a red neuronal e interpolación para el caso de la trayectoria de Salto A.

Algoritmo	Tiempo	
Red Neuronal	Entrenamiento: 7.857 s	Predicción: 6.307 s
	Total: 14.164 segundos	
Mínimos Cuadrados	Total: 0.387 segundos	

El algoritmo de red neuronal consta de dos pasos los cuales son entrenamiento y predicción. Se puede observar en la tabla 3.10 cómo el tiempo de ejecución de la red neuronal es considerablemente mayor que el tiempo de ejecución del algoritmo de interpolación, siendo la diferencia de aproximadamente 13.777 segundos, lo cual en términos de porcentaje significa un aumento en duración de aproximadamente 3660%. Tomando en cuenta que lo que aquí se pretende es el análisis de varias trayectorias para compararlas, esta diferencia de tiempo podría suponer una carga cuantiosa para el proceso total. Otro aspecto a notar es que la red neuronal necesita una cantidad considerablemente mayor de parámetros de sintonización para lograr los mejores resultados posibles, esto agregaría también tiempo al proceso de ejecución de este algoritmo. Los resultados presentados en la tabla 3.9 y la tabla 3.10 muestran que el algoritmo que más se adecúa para esta aplicación en concreto es el algoritmo de mínimos cuadrados, puesto que éste tuvo mejores resultados tanto en el ajuste de las trayectorias al tener un menor error en la mayoría de estas, como en el tiempo de ejecución, además de ser notablemente más sencillo de implementar, debido a que requiere un menor número de líneas de código y menor pre-procesamiento de los datos, por lo tanto es el algoritmo elegido para este trabajo.

Capítulo 4

Efecto de la iluminación en la definición de curvas de trayectoria y extracción de parámetros de similitud para comparación entre trayectorias

En este capítulo se presentan algunas pruebas realizadas para determinar cómo la variación en la iluminación del entorno del objeto estudiado afecta los resultados de la obtención de trayectorias. También se describe la extracción de los parámetros para comparación de similitud de trayectorias.

4.1 Determinación de la importancia de una iluminación constante en la captura de videos

Para las pruebas de iluminación se utilizó el motor con movimiento de vaivén. Las tomas de los videos se hicieron con ayuda de una mesa especial marca Polaroid MP4, la cual se muestra en la figura 4.1(a).

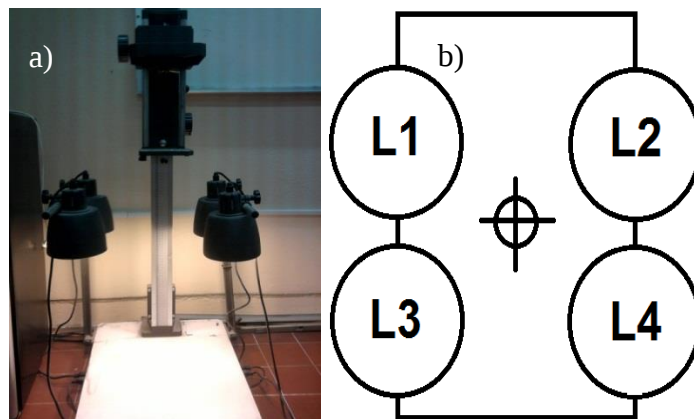


Figura 4.1. a) Mesa “Polaroid MP4 Land Camera”, cuenta con cuatro lámparas y altura ajustable para ubicar una cámara. b) Esquema simple de mesa polaroid MP4 vista desde arriba. L1 a L4 representa la posición física de cada lámpara que contiene la mesa.

Herramientas para determinación de regularidad de movimiento articulado mediante el seguimiento en una secuencia de imágenes.

Esta mesa brinda la facilidad de posicionar la cámara en el centro y ajustarle altura y ángulo, además de que incluye cuatro lámparas de tungsteno, las cuales se pueden fijar de manera que sean equidistantes al centro de la mesa. En la figura 4.1(b) se muestra un esquema simple de la mesa y la configuración de las lámparas, la marca del centro es donde se ubicó el motor para la toma de los videos. Las pruebas consistieron en determinar si los cambios en la iluminación del entorno pueden afectar los resultados de estimación de trayectorias de objetos; para esto se midió la iluminación a diferentes combinaciones de lámparas apagadas y encendidas, y se tomaron distintos videos para dichas combinaciones.

4.1.1. Medición de la intensidad de iluminación del entorno de trabajo

Para la medición de la intensidad de iluminación se utilizó un luxómetro marca Lutron, modelo Lx-1108, el cual se muestra en la figura 4.2(a). Este instrumento consiste de dos componentes, un sensor de iluminación o celda fotoeléctrica y una parte de control y pantalla para mostrar los valores de iluminación correspondientes a la zona medida. Estos dos componentes están separados por un cable para mayor versatilidad en la medición y lectura de datos por parte del operador. El sensor se colocó en el centro de la mesa Polaroid para medir la iluminación a distintas combinaciones de lámparas encendidas y apagadas, como se muestra en la figura 4.2(b).

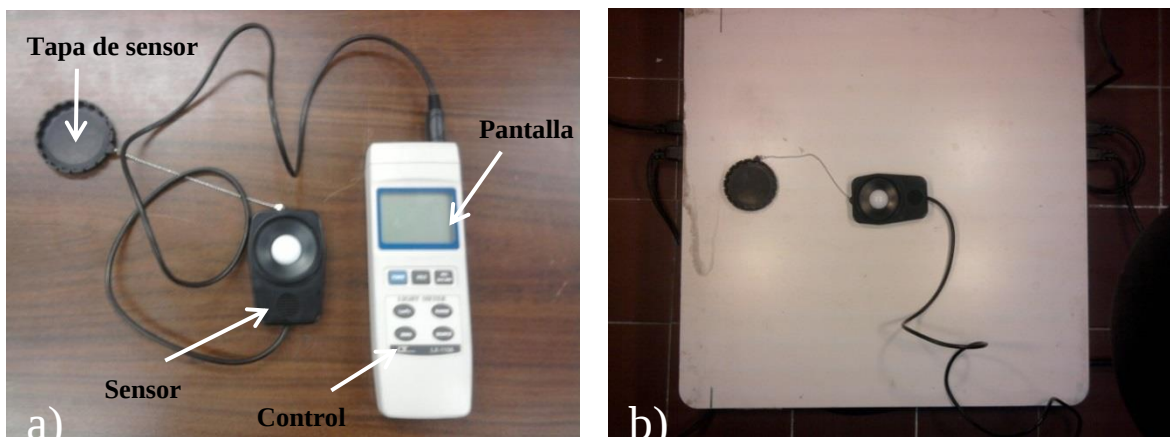


Figura 4.2. a) Luxómetro usado en pruebas de iluminación. b) Celda fotoeléctrica colocada en el centro de la mesa Polaroid para medición de iluminación.

Capítulo 4: Efecto de la iluminación en la definición de curvas de trayectoria y extracción de parámetros de similitud para comparación entre trayectorias

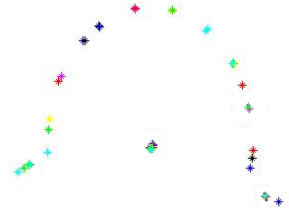
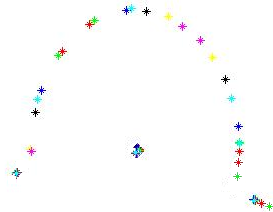
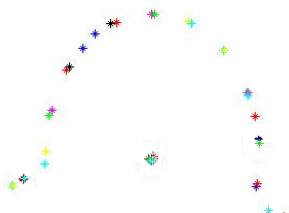
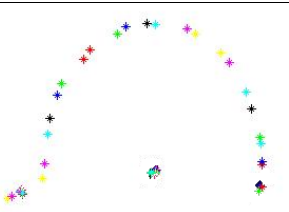
El procedimiento seguido para llevar a cabo las mediciones de intensidad de iluminación consistió en encender las cuatro lámparas y tomar la primera medición, a continuación solo se dejaron encendidas las dos lámparas del lado izquierdo (L1 y L3) y se tomó la siguiente medición, y así sucesivamente para las dos lámparas de la derecha encendidas (L2 y L4), para las cuatro lámparas apagadas, las dos lámparas de la parte baja encendidas (L3 y L4), las dos lámparas de la parte alta encendidas (L1 y L2), y apagando cada una de las lámparas por turno para tener las otras tres encendidas, lo cual generó cuatro videos más.

4.1.2. Resultados de las mediciones y trayectorias a diferentes grados de iluminación

En la Tabla 4.1 se muestran las trayectorias obtenidas para algunas combinaciones representativas de lámparas encendidas y las diferentes intensidades de iluminación medidas.

Herramientas para determinación de regularidad de movimiento articulado mediante el seguimiento en una secuencia de imágenes.

Tabla 4.1. Mediciones de iluminación y gráfica de trayectorias para distintas combinaciones de lámparas encendidas y apagadas.

Lámparas encendidas.	Valor de la Iluminación.	Trayectoria del motor de la figura 3.3
Todas	1911 Luxes	
L2 y L4	1028 Luxes	
L1 y L3	1037 Luxes	
Ninguna	215.4 Luxes	

Las trayectorias graficadas son muy similares a simple vista, sin embargo, para poder determinar si las trayectorias son diferentes, se necesita de un procedimiento más completo que la simple observación. Se utilizó el algoritmo basado en la red neuronal y el

Capítulo 4: Efecto de la iluminación en la definición de curvas de trayectoria y extracción de parámetros de similitud para comparación entre trayectorias

error cuadrático medio una comparación más precisa de las trayectorias mostradas en la tabla 4.1. El procedimiento de comparación es el siguiente: Una vez obtenidas las trayectorias para todas las combinaciones de lámparas encendidas y apagadas que se desean comparar, se tomó como referencia la trayectoria de todas las lámparas encendidas. La trayectoria de referencia se sometió al algoritmo de red neuronal para su entrenamiento, y se creó la función matemática que describe el comportamiento de esta trayectoria, la cual se muestra en la figura 4.3(a), donde se observan tanto los puntos como la curva de trayectoria ajustada. Para comparar la trayectoria de referencia con las demás trayectorias, se obtuvo el ECM entre ellas. En la figura 4.3(b-f) se muestran las comparaciones entre la curva de la trayectoria de referencia ajustada (línea azul) y los puntos de las trayectorias con las cuales se hicieron estas comparaciones. Se puede observar como los puntos de las trayectorias comparadas se desvían de la línea azul (trayectoria de referencia), indicando que la percepción de una trayectoria por parte de la cámara puede variar para diferentes patrones de iluminación. Para tener una comparación justa, todos los videos se dividieron en la misma cantidad de imágenes (50 imágenes), también se procuró que todos los videos se tomaran en la misma posición de inicio.

Herramientas para determinación de regularidad de movimiento articulado mediante el seguimiento en una secuencia de imágenes.

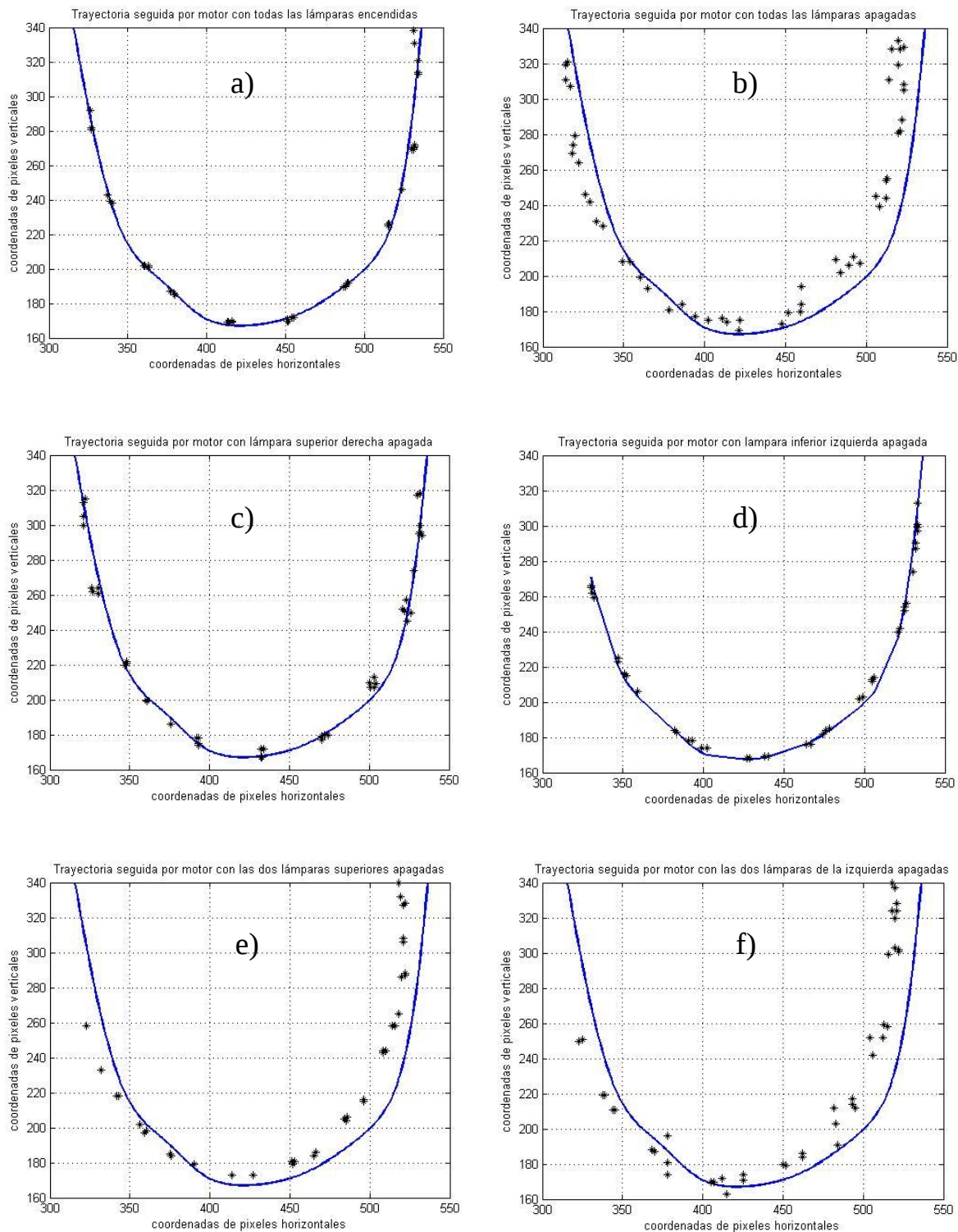


Figura 4.3. Trayectoria del motor tipo péndulo y su comparación consigo misma para diferentes configuraciones de iluminación. a) Todas las lámparas encendidas, b) Todas las lámparas apagadas,

Capítulo 4: Efecto de la iluminación en la definición de curvas de trayectoria y extracción de parámetros de similitud para comparación entre trayectorias

c) Lámpara L2 apagada, d) Lámpara L1 apagada, e) Lámparas L1 y L2 apagadas, f) Lámparas L2 y L4 apagadas

En la tabla 4.2 se pueden observar los distintos ECM correspondientes a las comparaciones realizadas entre la trayectoria de referencia con respecto a distintas combinaciones de lámparas. Se puede ver que cuando la iluminación varía, la trayectoria se percibe de manera diferente aun cuando la cámara y los objetos a seguir no cambien de posición.

Lámparas encendidas	Error cuadrático medio
Todas (referencia)	57.7
Ninguna	476.0
L1, L3 y L4	96.9
L1, L2 y L4	127.4
L3 y L4	771.8
L2 y L4	1568.4

Tabla 4.2. Error cuadrático medio al comparar trayectoria de referencia con distintos patrones de iluminación para videos de motor con movimiento de vaivén.

4.2 Análisis del error de captura de centroides en videos de motor de vaivén

Durante las pruebas se observó que, aunque el *target* ubicado en el centro del mecanismo de motor de vaivén solo tiene movimiento de rotación, el centroide aparece desplazado para diferentes imágenes del video, de manera que, para distintas imágenes, el centroide se presenta en distintas coordenadas, esto puede ser debido a vibraciones,

Herramientas para determinación de regularidad de movimiento articulado mediante el seguimiento en una secuencia de imágenes.

percepción de la cámara utilizada, así como la forma del *target* colocado, la cual no fue totalmente redonda. En la figura 4.4 se aprecia un acercamiento a los centroides obtenidos para el *target* central en el video de todas las lámparas encendidas.

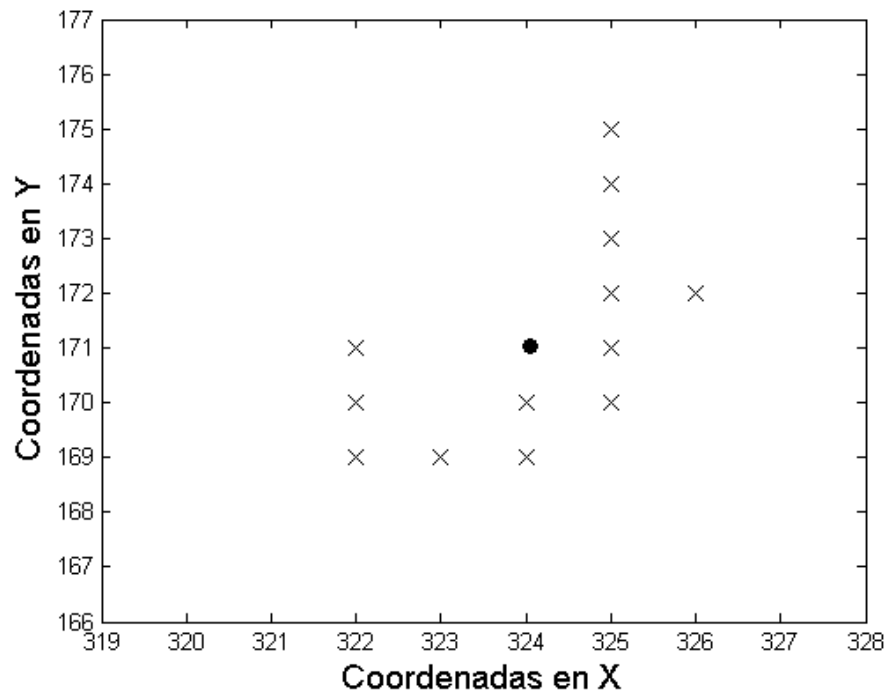


Figura 4.4. Acercamiento a los centroides correspondientes al *target* central del mecanismo de motor de vaivén en el video de todas las lámparas encendidas, el círculo negro es la media de los datos.

Para obtener el error de captura de centroides, se obtuvo la tabla mostrada en el apéndice C.2 esta tabla contiene las coordenadas de los centroides de la figura 4.4 y se obtuvo la media de las coordenadas la cual fue de 324.0577 en X y 171.0385 en Y, la desviación estándar fue de ± 1.3048 en X y de ± 1.95 en Y, estos últimos valores se tomaron como error de captura.

4.3 Extracción de parámetros de similitud de curvas de trayectorias

En esta sección se describe la extracción de los parámetros para la comparación de similitud de trayectorias.

4.3.1. Descripción de parámetros de similitud de trayectorias

A continuación se muestra una lista con los parámetros aquí planteados para la comparación de similitud de trayectorias, algunos de los cuales son tomados de otros trabajos como [15], [16], [22] y [26], mientras que otros son propuestos en el presente trabajo, en conjunto se forma un grupo más amplio de variables de donde seleccionar para clasificación y comparación de trayectorias. A continuación se muestra una lista de los parámetros de similitud utilizados y sus definiciones.

- **Rectitud (ST):** Medida de la desviación que existe en la trayectoria con respecto a una línea recta trazada del punto inicial al punto final de la misma.

$$ST = \frac{dE}{L} \quad (8.1)$$

Dónde:

- ST : Índice de rectitud.
- dE : Distancia euclidiana entre el punto inicial y final de la trayectoria.
- L : Longitud total de la trayectoria.

El parámetro de rectitud se puede obtener gracias a la función **rectitud_AI**, la cual se muestra en el apéndice B.12.

- **Duración (DT):** Duración total de la ejecución de la trayectoria desde su inicio hasta su fin tomando en cuenta también su punto de inicio y final.

Herramientas para determinación de regularidad de movimiento articulado mediante el seguimiento en una secuencia de imágenes.

$$DT = (f(final) - f(inicial)) * fps \quad (8.2)$$

Dónde:

- $f(final)$: Número de cuadro final de la trayectoria.
- $f(inicial)$: Número de cuadro inicial de la trayectoria.
- fps : Cuadros por segundo del video del cual se extrajo la trayectoria.

- **Velocidad (VT):** Velocidad del objeto

$$VT = \sum_{i=1}^n \sqrt{(X_i - X_{i-1})^2 + (Y_i - Y_{i-1})^2} / fps \quad (8.3)$$

Dónde:

- n : Total de cuadros que forman la trayectoria.
- i : Cuadro en turno.
- (X,Y) : Coordenadas de los puntos de la trayectoria.
- fps : Cuadros por segundo del video del cual se extrajo la trayectoria.

Este parámetro puede obtenerse utilizando la función **velocidad_AI**, esta función se puede observar en el apéndice B.11.

- **Puntos de inflexión de la trayectoria:** Puntos de la trayectoria donde sus valores en X pasan de un tipo de concavidad a otra. Puntos donde $f''(X) = 0$, el algoritmo utilizado para encontrar este parámetro se llama **pInflexion_AI** y se puede observar en el apéndice B.10.
- **Extremos:** Valores más grandes (máximos) o más pequeños (mínimos) que toma la trayectoria ya sea dentro de una región particular (locales), o en el dominio de la función en su totalidad (globales).
 - Si $f'(X) = 0$ y el intervalo de la trayectoria es cóncava hacia abajo ($f''(X) < 0$) se trata de un máximo.

Capítulo 6: Extracción de parámetros de comparación de trayectorias

- Si $f'(X) = 0$ y el intervalo de la trayectoria es cóncava hacia arriba ($f''(X) > 0$) se trata de un mínimo.

Estos parámetros se obtienen con el algoritmo **pCríticos_AI**, el cual se muestra en el apéndice B.9.

- **Intervalos de crecimiento y decrecimiento:** Intervalos en los cuales se presentan incrementos o decrementos de la trayectoria en el valor de Y con respecto a X.
- **Intervalos de concavidad:** Intervalos de inicio y fin de las concavidades presentadas en la trayectoria.
- **Error Cuadrático Medio:** Diferencia entre dos trayectorias punto a punto comparando sus coordenadas en (X,Y), este dato puede ser obtenido mediante la función **ecm_AI**, el cual se muestra en el apéndice B.7.
- **Correlación Cruzada (T1 * T2):** Medida cuantitativa de similitud entre dos funciones o señales [35], este parámetro se obtiene con la función **xcorr cov_AI**, la cual se muestra en el apéndice B.8
- **Covarianza Cruzada:** Tiene la misma definición que la correlación cruzada, con la única diferencia que a las señales a comparar se les subtrae la media como una forma de normalización [35], este parámetro se obtiene con la función **xcorr cov_AI**, la cual se muestra en el apéndice B.8
- **Ángulo de dirección del objeto:** Se refiere al caso de que el objeto de interés sea una persona, y trata de relacionar los planos imaginarios que dividen al cuerpo en secciones para la descripción y ubicación de sus partes [36]. En la figura 4.5 se pueden observar los planos anatómicos estándar. El ángulo de dirección de una persona es la intersección del plano coronal y el plano sagital con respecto a la horizontal del video.

Herramientas para determinación de regularidad de movimiento articulado mediante el seguimiento en una secuencia de imágenes.

Hasta el momento no se ha encontrado otra aplicación que implemente este tipo de algoritmo para esta aplicación en particular.

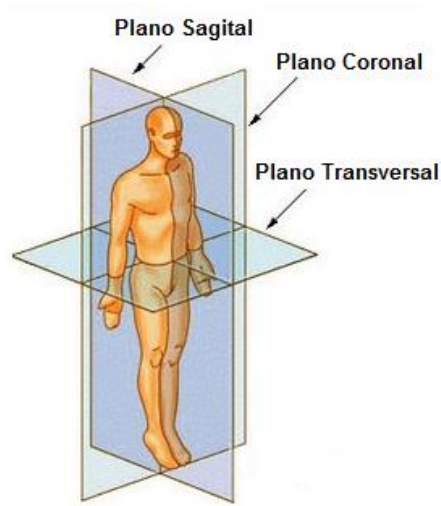


Figura 4.5. Planos anatómicos humanos (Imagen extraída de Moore L., “Anatomía con orientación clínica”, [36]).

4.3.2. Obtención del parámetro de ángulo de dirección de objeto

Este parámetro es más laborioso que los anteriores, a continuación se presenta como llevar a cabo el proceso de su obtención. La función para realizar este proceso se puede observar en el apéndice B.13 y se le nombró “**angDir_AI**”. Para la obtención del parámetro de ángulo de dirección de objeto es necesario cargar el video a analizar y los centroides de las trayectorias obtenidos anteriormente con sus respectivos índices de cuadro **i**. A continuación se establecen los cuadros inicial y final de análisis, y también se establece el intervalo de sub-muestreo. El algoritmo basa su funcionamiento en el método de substracción de fondo (apéndice A.3.2) y consiste en la creación de un fondo general mediante el promediado de todas las imágenes del video; este fondo servirá para distinguir cuales son los pixeles en los que ocurrió un movimiento (cambio) dentro de una imagen. Las imágenes de las cuales se ha de restar el fondo creado han de ser pre-procesadas, siendo estas convertidas a escala de grises y filtradas con un filtro de mediana para reducir

efectos de ruido. Para hacer la substracción de fondo y el procedimiento subsecuente, se utiliza un ciclo que va adquiriendo las imágenes del video a los intervalos de sub-muestreo establecidos, el algoritmo para substracción de fondo se nombró **subsFondo_AI** y se muestra en el apéndice B.14. Para que el procedimiento sea más rápido, las imágenes así como el fondo son recortados a la zona donde se ubicó el centroide (región de interés); este recorte o región de interés se consideró de 30x30 píxeles para garantizar que contenga a la gimnasta pero sin consumir espacio innecesario. También, el algoritmo funciona de manera que, de haber una imagen en la cual no se detectó centroide (por ejemplo un falso negativo), esa imagen no será procesada y se pasará a la siguiente. En la figura 4.6(a) se puede observar la región de interés alrededor del centroide detectado en esa imagen; mientras que la figura 4.6(b) corresponde al fondo, por su parte en la figura 4.6(c) se muestra la diferencia entre las figura 4.6 (a) y (b).

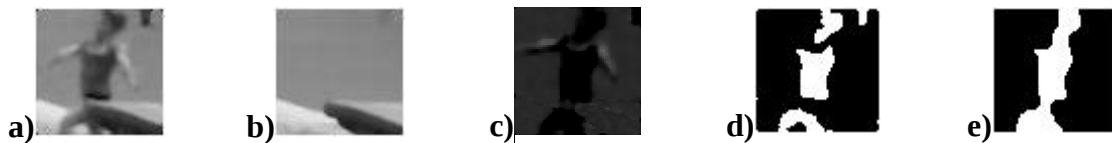


Figura 4.6. a) Imagen pre procesada. b) Imagen con fondo recortado. c) Imagen con substracción de fondo. d) Detección de gimnasta mediante algoritmo de Ridler Calvard. e) Imagen de gimnasta después de ser procesada.

La imagen resultante de la substracción de fondo se somete a un algoritmo llamado Ridler Calvard. Este método consiste en un algoritmo iterativo que estima un umbral óptimo que separe a un objeto del fondo de acuerdo a las medias de los niveles de gris de la imagen [23]. El algoritmo inicia obteniendo la media total de las intensidades de los píxeles de la imagen de la figura 4.6(c), ésta media la utiliza como valor de umbral **T** para generar una imagen binaria **Y**, donde los píxeles cuya intensidad sea mayor al umbral **T** tendrán un valor de 1 mientras que los píxeles cuya intensidad sea menor a este umbral tendrán un valor de 0. Teniendo esta imagen binaria, el algoritmo entra en un ciclo en el cual se obtienen dos medias μ_1 y μ_2 , donde μ_1 es la media de todos los valores de la imagen en escala de grises de la figura 4.6(c) cuyas posiciones corresponden a las posiciones de valor 1 de la imagen binaria **Y**, mientras que μ_2 es la media correspondiente a las posiciones de valor 0. Teniendo μ_1 y μ_2 , se actualiza el umbral **T** con la siguiente operación:

Herramientas para determinación de regularidad de movimiento articulado mediante el seguimiento en una secuencia de imágenes.

$$T = (\mu_1 + \mu_2)/2$$

A continuación se actualiza la imagen binaria **Y** con el umbral **T** actualizado. De nuevo se obtienen dos medias μ_1 y μ_2 , y se actualiza **T** y el proceso se repite **k** veces hasta lograr que el umbral **T** que separa al objeto de interés del fondo de la imagen no varíe significativamente; el número de ciclos **k** encontrado que logró buenos resultados fue de **k=25**. El algoritmo de Ridler Calvard, el cual se muestra en el apéndice B.15 y fue nombrado **rCalvard_AI**, genera una imagen binaria con dos regiones, una en blanco (valores en 1's) y otra en negro (valores en 0's), el algoritmo no distingue cual color es el objeto y cuál es el fondo, por lo que se tuvo que hacer esta distinción mediante un comando de decisión, tomando en cuenta que el tamaño de fondo siempre será mayor al tamaño del objeto. Se puede apreciar en la figura 4.6(d) la imagen con la gimnasta detectada dentro de un fondo negro. Al igual que con las implementaciones de obtención de trayectorias mostradas en la sección 3.3 surgen problemas de aparición de huecos dentro del objeto, por lo que de nuevo se deben tomar medidas para rellenar el objeto detectado y eliminar todos aquellos pixeles que no son parte del objeto mediante operaciones morfológicas como dilatación, erosión y cerradura. Una vez llevado a cabo este proceso, la imagen queda como en la figura 4.6(e).

Teniendo la figura 4.6(e), se pueden obtener las direcciones en las cuales apunta el objeto aplicando un procedimiento llamado Análisis de Componentes Principales (PCA). El algoritmo de PCA implementado busca la dirección en la cual se encuentra una cantidad mayor de pixeles pertenecientes al objeto detectado, en este caso la dirección es la que apunta desde la cabeza hacia los pies de la gimnasta, a esta dirección se le conoce como el primer componente principal, mientras que la perpendicular a esta dirección sería el segundo componente principal. En la figura 4.7 se muestra un ejemplo con los componentes principales encontrados.



Figura 4.7. Imagen de ejemplo con componentes principales encontrados en la gimnasta.

Una vez que se tienen las direcciones de los componentes principales del objeto (gimnasta), se pueden obtener los ángulos de estas; para hacer comparaciones entre las distintas trayectorias, se optó por obtener el ángulo que existe entre la horizontal del video y la línea del primer componente principal que va del centroide de la gimnasta a su cabeza, como se observa en la figura 4.8.



Figura 4.8. Ilustración del ángulo de dirección de objeto.

Pero el algoritmo de PCA tiene el inconveniente de que encuentra las direcciones mas no identifica cuál de los extremos apunta hacia la cabeza y cual hacia los pies, esto ocasiona el problema de que, para cada posición de la gimnasta, se tendrán dos ángulos posibles para el primer componente principal, un ángulo que apunta hacia la cabeza y otro hacia los pies.

Herramientas para determinación de regularidad de movimiento articulado mediante el seguimiento en una secuencia de imágenes.

Por ejemplo, cuando la gimnasta está totalmente de pie, el algoritmo de PCA no distingue si se tiene un ángulo de 90° o de 270° , de igual manera para el segundo componente principal se tendrán dos ángulos posibles. Aún con la rotación de la gimnasta, se desea conocer la posición de la cabeza en todo momento. Para solucionar este problema, se tomó como referencia y se capturó el ángulo que apunta hacia la cabeza de la gimnasta en la primera imagen analizada, a partir de este paso, cada ángulo de dirección se calculó tomando en cuenta el ángulo de la imagen anterior utilizando sentencias de condición y obteniendo la diferencia de los dos ángulos posibles de la imagen presente con respecto al de la imagen anterior para relacionar cuál de los dos es más cercano. El extremo que proporcione un ángulo con menor diferencia con respecto al cuadro anterior será el que corresponde con el que apunta hacia la cabeza en la imagen anterior. Los ángulos de dirección se calcularon utilizando operaciones con funciones trigonométricas y tomando en cuenta las coordenadas del centroide y el punto final del primer componente principal.

En la figura 4.9 se muestra el resultado de la obtención de los dos componentes principales para una serie de cuadros de los videos de la gimnasta.

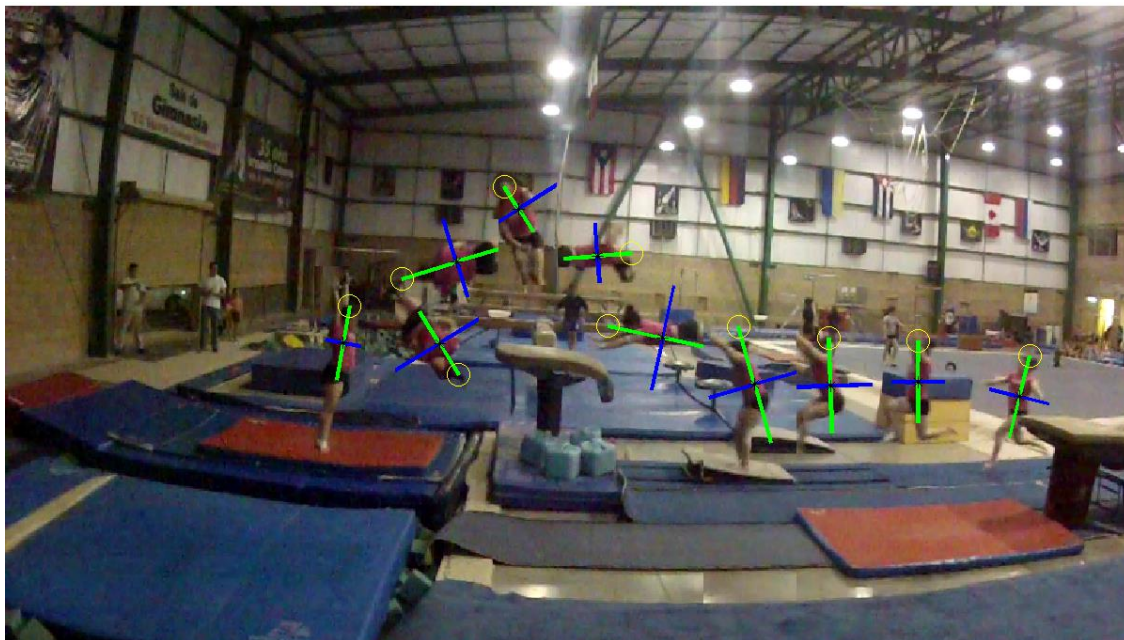


Figura 4.9. Ejemplo de obtención de ángulos de dirección de objeto para una serie de cuadros de video de gimnasta.

Capítulo 6: Extracción de parámetros de comparación de trayectorias

A continuación se presenta un algoritmo en pseudocódigo para un mejor entendimiento del proceso.

```
Video = Leer video; intervalo de sub muestreo = leer intervalo de sub
muestreo;

Centroides = leer centroides;

For i = imagen inicial: intervalo de sub muestreo: imagen final
    if (existe al menos un centroide en imagen en turno)
        Recorta imagen y fondo a una zona alrededor del centroide;

        Validación de recorte para evitar salir de los límites de la
        imagen;

        Pre procesamiento de imagen en turno (Filtrado de mediana y
        conversión a escala de grises);

        Substracción de fondo, imagen_Substraída= imagen - fondo;

        Algoritmo de Ridler Calvard para umbralado y creación de
        imagen binaria;

        Procesado de imagen binaria (Dilatación, erosión y
        cerradura);

        Determinación de componentes principales (PCA);

        Cálculo de ángulos de dirección utilizando sentencias de
        decisión que toman en cuenta el ángulo de la imagen anterior
        para decidir cuál de los dos ángulos posibles apunta hacia la
        cabeza de la gimnasta.

    Else (No existe ningún centroide en imagen en turno)
        Ignora imagen en turno y continúa con imagen siguiente;

    End
End
```

End

Herramientas para determinación de regularidad de movimiento articulado mediante el seguimiento en una secuencia de imágenes.

4.3.3. Comparación de parámetros de similitud para videos de saltos de gimnasta

En el apéndice B se muestran los algoritmos creados para los parámetros de similitud mostrados en la sección 4.3.1, explicando cada paso tomado en ellos, las funciones han sido probadas para todos los videos de péndulo y gimnasta y funcionan correctamente. Una vez extraídos los parámetros de similitud, se hace una comparación entre estos parámetros para distintas trayectorias para determinar el grado de similitud que existe entre ellas. En este caso, la comparación de parámetros se hace entre las trayectorias de los videos de saltos. Como muestra de comparación, en la figura 4.10 se pueden observar los parámetros de máximos, mínimos y puntos de inflexión graficados para las trayectorias de los tres videos de la gimnasta.

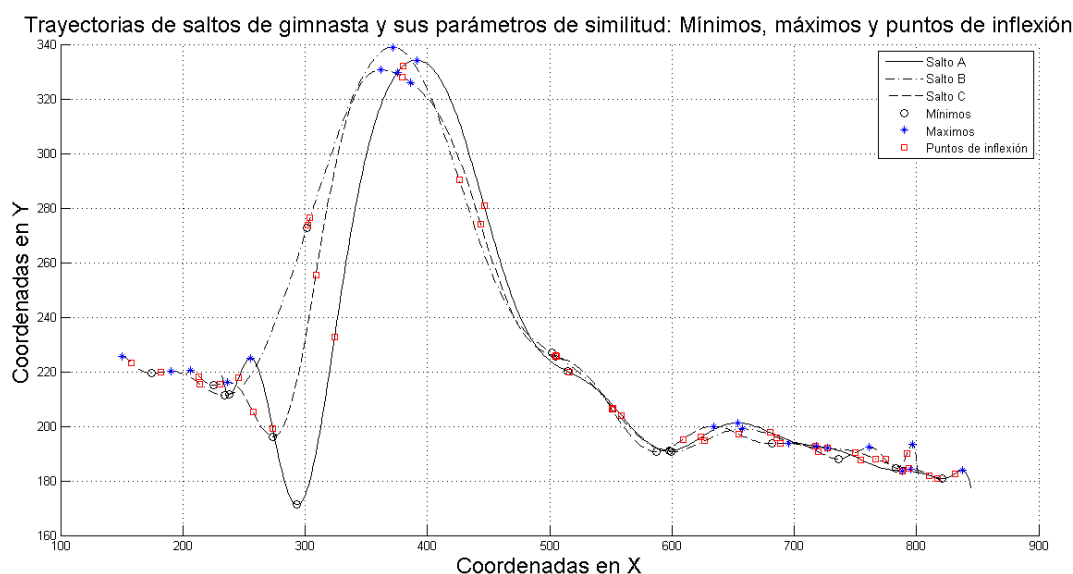


Figura 4.10. Trayectorias de salto de gimnasta con parámetros de similitud de mínimos (círculos negros), máximos (asteriscos azules) y puntos de inflexión (cuadros rojos).

Se puede ver por ejemplo que en una de las trayectorias se tiene un máximo de un valor mayor que en las otras dos, mientras que en una de las trayectorias se puede ver una caída más prolongada y que llega a un punto inferior a las otras dos; este tipo de comparación se puede utilizar para encontrar la similitud. En la tabla 4.3 se observan las diferencias entre

Capítulo 6: Extracción de parámetros de comparación de trayectorias

los parámetros de similitud planteados en la sección 4.3.1, las comparaciones fueron realizadas para los videos de la gimnasta. La primera columna de la tabla 4.3 corresponde al nombre del parámetro comparado, la segunda corresponde a la diferencia entre los saltos A y B, mientras que la tercera columna corresponde a la diferencia de los parámetros entre los saltos A y C.

Tabla 4.3. Comparación de parámetros de similitud para distintos saltos.

Nombre del Parámetro de Comparación de Similitud.	Saltos A y B	Saltos A y C
Error Cuadrático Medio (ECM)	3.0861e+10	1.6336e+07
Correlación cruzada (XCORR)	0.9631	0.9923
Covarianza cruzada (XCOV)	0.6516	0.8756
Índice de rectitud (ST)	0.1256	0.1966
Diferencia máximo absoluto (DIFMAX)	4.6932	3.5689
Diferencia mínimo más bajo al caer (DIFMIN)	40.1105	24.9418
Diferencia velocidad promedio (DIFVEL)	182.9551	151.1258
Diferencia en rebote (DIFREB)	44.7959	29.4404
Diferencia promedio ángulo de dirección de la gimnasta (DIFANG)	18.2392	0.5625

Herramientas para determinación de regularidad de movimiento articulado mediante el seguimiento en una secuencia de imágenes.

Para los parámetros de similitud mostrados en la tabla 4.3, a excepción de la correlación cruzada y la covarianza cruzada, entre mayor sea el valor del parámetro, menor es la similitud entre las dos trayectorias comparadas. La correlación cruzada y covarianza cruzada están en el rango [0,1], y entre más se acerque el índice a 1, más similares son las trayectorias comparadas. Es notable como en todos los parámetros de similitud planteados, a excepción del índice de rectitud (ST) tienen valores que indican que las trayectorias A y C son más similares entre sí en comparación con las trayectorias A y B tal como debe ser, puesto que las trayectorias A y C corresponden al mismo tipo de salto mientras que la trayectoria B corresponde a un salto distinto.

4.3.4. Parámetro de evaluación para determinación de similitud entre trayectorias

Para utilizar los parámetros de similitud, se ha de crear un vector **P** de parámetros por cada par de trayectorias que se deseen comparar; cada comparación poseerá su vector **P**, por lo que si es diferente y única podrá ser distinguida de las demás gracias a este vector. También se crea un vector de pesos **W**, este vector asignará un peso a cada uno de los parámetros de similitud de acuerdo a su grado de importancia. Los pesos dependerán de la aplicación (trayectoria de la actividad realizada por el objeto) y de la experiencia de la persona que los asigne. Por ejemplo, en el caso de la gimnasta, un entrenador calificado deberá decidir qué parámetros de la trayectoria tendrían una mayor importancia para calificarla como regular o irregular, o como un salto bueno o malo (por ejemplo, una altura más elevada, un rebote menos pronunciado, etc.). Por lo tanto se tendrán dos vectores, el vector **P** del valor de los parámetros de comparación y el vector **W** de asignación de pesos según la importancia de cada parámetro:

$$\mathbf{P} = [p_1, p_2, p_3, \dots, p_m]$$

$$\mathbf{W} = [w_1, w_2, w_3, \dots, w_m]$$

Capítulo 6: Extracción de parámetros de comparación de trayectorias

Donde m es el número total de parámetros extraídos de las trayectorias, $p_1, p_2, \dots$ son los parámetros de comparación para evaluar regularidad y $w_1, w_2, \dots$ son los pesos asignados a cada parámetros según su importancia. Entonces, se define una variable ϵ , la cual servirá como parámetro de evaluación, mediante la ecuación (4.1):

$$\epsilon = W * P \quad (4.1)$$

Donde ϵ es inversamente proporcional a la similitud que existe entre dos trayectorias, es decir, al comparar dos trayectorias, entre menor sea ϵ , mayor es la similitud que existe entre ellas. A modo de ejemplo, utilizando los valores de los parámetros de similitud mostrados en la tabla 4.3 para los videos de la gimnasta, se tiene el siguiente vector P :

$$P = [ECM, \frac{1}{xcorr}, \frac{1}{xcov}, rectitud, difMax, difMin, difVel, difReb, difAng]$$

Sustituyendo los valores obtenidos para los parámetros de similitud de la tabla 4.3, se forman dos vectores de comparación P_{AB} y P_{AC} , donde el vector P_{AB} es el resultado de sustituir los valores de comparación de los parámetros de similitud ente las trayectorias A y B, mientras que el vector P_{AC} corresponde a los parámetros de similitud en la comparación entre las trayectorias A y C. Para hacer la evaluación se crea un vector de pesos W que garantice que ninguno de los parámetros de similitud de los vectores P_{AB} y P_{AC} domine sobre los demás. De modo que se tienen tres vectores:

$$P_{AB} = [3.086e + 10, 0.963, 0.651, 0.125, 4.693, 40.110, 182.955, 44.795, 18.239]$$

$$P_{AC} = [1.633e + 07, 0.992, 0.875, 0.196, 3.568, 24.941, 151.125, 29.440, 0.562]$$

$$W = [1e - 08, 10, 10, 10, 1, 1, 1, 1, 1]$$

En P , se establecen los valores en el mismo orden que en la tabla 4.3, hay que recordar que en el caso de los parámetros de correlación cruzada y covarianza cruzada, un mayor valor de estos parámetros indica una mayor similitud entre las trayectorias, es por ello que estos dos parámetros se deben modificar obteniendo su inverso ($1/xcorr, 1/$

Herramientas para determinación de regularidad de movimiento articulado mediante el seguimiento en una secuencia de imágenes.

xcov). En este caso el vector **W** se eligió de una forma conveniente para dar un ejemplo claro de cómo llevar a cabo el proceso de comparación de trayectorias, los valores de este vector se eligieron de manera que ningún parámetro domine sobre los demás, dándoles una ganancia que incremente aquellos parámetros que tienen valores pequeños (rectitud, correlación cruzada y covarianza cruzada) y una atenuación al parámetro de ECM, puesto que este valor es muy alto comparado con los demás. Sustituyendo los valores de P_{AB} , P_{AC} y **W** en la ecuación (4.1) se obtienen los resultados de la tabla 4.4. Estos resultados indican que las trayectorias de los saltos A y C son más similares entre sí que las trayectorias pertenecientes a los saltos A y B. A partir de este punto, se puede establecer un valor umbral del parámetro de evaluación ϵ que especifique cuando una trayectoria puede ser considerada como regular o irregular con respecto a otra trayectoria de referencia.

Tabla 4.4. Parámetro de evaluación ϵ para las comparaciones entre las trayectorias de gimnasta.

Trayectorias comparadas	Parámetro de evaluación (ϵ)
A y B	616.7820
A y C	230.4293

Capítulo 5

Discusión, conclusiones y trabajos futuros

En este capítulo se hace una discusión acerca de los resultados de la investigación. Después se abordan una serie de conclusiones obtenidas del desarrollo de la tesis. Para finalizar se plantean los trabajos, pruebas e implementaciones futuras que se pueden realizar tanto para optimización de las implementaciones y algoritmos creados, como para el aprovechamiento de este trabajo de tesis.

5.1 Discusión y conclusiones

Los algoritmos descritos sobre la obtención de trayectorias toman como base los presentados en [33], pero estos se mejoraron, de manera que se tiene un sistema más eficiente con menor tiempo de procesamiento y mayor flexibilidad de ajustes al algoritmo. Se aprovecha el control de flujo indexado para facilitar la prueba con distintos videos, teniendo una reducción de líneas de código de aproximadamente cuatro veces. Las trayectorias obtenidas mediante los algoritmos aquí desarrollados tendrán, en el mejor de los casos, una precisión que dependerá de la resolución de la imagen y por lo tanto de la cámara utilizada en las tomas de los videos.

5.1.1 Conclusión sobre la técnica seleccionada para ajustes de curva de trayectoria

En el caso de las técnicas utilizadas para ajuste de trayectorias, las cuales fueron red neuronal y mínimos cuadrados. Teniendo en cuenta las pruebas realizadas con estas dos técnicas y los resultados mostrados en la tabla 3.9 y la tabla 3.10 (Ver Capítulo 3), el algoritmo que más se adecuaba para esta aplicación es el algoritmo de mínimos cuadrados,

Herramientas para determinación de regularidad de movimiento articulado mediante el seguimiento en una secuencia de imágenes.

puesto que éste tuvo mejores resultados tanto en el ajuste de las trayectorias al tener un menor error en la mayoría de estas, como en el tiempo de ejecución. Además el método de mínimos cuadrados es notablemente más sencillo de implementar, debido a que requiere un menor número de líneas de código y un menor pre-procesamiento de los datos, por lo tanto es el algoritmo elegido para este trabajo.

5.1.2 Conclusión sobre la obtención del parámetro de evaluación de similitud ϵ

El parámetro de evaluación ϵ tuvo valores congruentes con las trayectorias de saltos comparadas. Se puede observar en la tabla 4.4 como el parámetro de evaluación ϵ entre las trayectorias A y C es menor al parámetro para las trayectorias A y B, tal como debe ser, pues esto indica que la trayectoria A es más similar con respecto a C que con respecto a B. En este caso no se tiene un umbral de regularidad, pero si se tiene un parámetro de evaluación ϵ que indica qué tanta es la diferencia entre una trayectoria y otra. Para poder establecer un umbral de regularidad se necesitaría contar con una persona experta en el tema de saltos de gimnasia que establezca cuales de los saltos capturados pueden ser considerados como regulares y cuáles no (Por ejemplo, considerar como salto regular a aquel que el experto señale como un buen salto, mientras que un mal salto sería considerado como irregular), a partir de este análisis se puede fijar un umbral que establezca que valor del parámetro de evaluación ϵ caería dentro de lo establecido como regular para futuras trayectorias.

5.1.3 Ventajas en el uso de las herramientas diseñadas en comparación con otros sistemas y/o métodos

En esta sección se muestran las ventajas que el uso de los algoritmos creados en esta tesis.

- **No se requiere de equipamiento especial**

Para llevar a cabo tareas de seguimiento de trayectorias, en muchas aplicaciones se requiere de equipamiento especializado, como los equipos de OptoTrak Certus, Vikon, etc., así como software especial, *targets* de detección particulares, etc. El adquirir un equipamiento de este tipo eleva bastante los costos de experimentación e implementación, además de que este tipo de equipo requiere mucho espacio o aspectos complicados de instalación. Con el software aquí diseñado, no se requiere equipamiento especial de ningún tipo, solamente se requiere de una computadora y una cámara de video convencional. En cualquier sistema de medición, una mejor infraestructura favorecerá la calidad de los resultados obtenidos. En cuanto a los *targets* de detección, no es necesario ningún tipo de conexión ni material especializado, puesto que los algoritmos funcionan con cualquier tipo de *target*, simplemente basta con lograr contrastar al objeto con el fondo de la imagen. Todo esto garantiza un costo mucho menor de implementación, con un error encontrado en la detección de los centroides de ± 1.3048 en X y de ± 1.95 en Y, en la sección 4.2 se explicó cómo se obtuvo este error.

- **Variedad de opciones de uso de lugar y aplicación**

En cuanto a la versatilidad de uso, si se compara lo aquí implementado con las tecnologías actuales, se tiene una mayor facilidad de aplicación, puesto que al requerirse menos equipamiento especial, se puede aplicar en cualquier medio sin traslados de material, y con menor riesgo de dañar algún aparato costoso. Pero se recomienda que el ambiente en el cual se ha de aplicar sea controlado en cuanto a iluminación y evitando que haya interferencia de colores con el *target*, no obstante, los algoritmos se probaron con videos donde existen ambos problemas de cambios de iluminación e interferencia de color y funcionaron de manera aceptable. Al haber sido diseñados como funciones tipo módulos, los algoritmos creados también presentan versatilidad en cuanto a su aplicación, es decir, pueden ser aplicables para seguimiento de objetos tan variados como los videos de gimnasta y los videos de motor tipo péndulo; basta con determinar cuáles de los parámetros

Herramientas para determinación de regularidad de movimiento articulado mediante el seguimiento en una secuencia de imágenes.

de similitud serían aplicables en una secuencia, para así poder utilizar solo aquellas funciones que se requieran.

5.1.4 Innovaciones y mejoras

Se puede comentar que hasta el momento, no se ha encontrado una aplicación que de una secuencia de video realice seguimiento de uno o más objetos de interés, genere sus trayectorias automáticamente y además extraiga parámetros de dichas trayectorias para su caracterización y comparación, todo en un mismo conjunto de herramientas. En muchos casos se utilizan algunos tipos de dispositivos de posicionamiento junto con base de datos de trayectoria. Aunque para la utilización del algoritmo aquí creado se necesita del establecimiento manual de un umbral para el color del *target*, todo el procedimiento subsiguiente se ejecuta de manera totalmente automática. Además, cabe mencionar que los algoritmos aquí desarrollados se han utilizado funciones comunes incluidas en Matlab.

En [16] se aborda la creación de una librería de extracción de parámetros de similitud de trayectorias, pero existen algunas diferencias entre esta librería y los algoritmos aquí realizados. En primer lugar, la librería creada en [16] no incluye la fase de obtención de las trayectorias, sino que se enfoca solamente a la extracción de parámetros con trayectorias ya dada. Los parámetros aquí extraídos así como la clasificación de estos difieren en gran medida de los tratados en [16], mientras que en el presente trabajo se extrae un total de nueve parámetros, en [16] se enfocan en extraer solo tres: Posición, velocidad y aceleración, a partir de estas tres características se derivan otras, algunos de los parámetros del presente trabajo se basaron en las ideas planteadas en esta referencia, aunque con modificaciones. Después de realizar una búsqueda exhaustiva, los únicos trabajos concernientes a la extracción de parámetros de similitud para clasificación de trayectorias que se encontraron fueron los de [15], [16], [22] y [26], nuestra investigación presenta un método novedoso con respecto al tema análisis de movimiento de objetos, seguimiento y análisis de trayectorias. Además, mientras que en [15] los análisis se presentan solo

teóricamente, en el presente trabajo se aborda también un análisis desde el punto de vista práctico, exponiendo un conjunto de algoritmos que representan un avance en investigaciones acerca del tema de seguimiento y análisis de trayectorias.

Entre los parámetros de similitud de trayectorias extraídos, se menciona la extracción del ángulo de dirección utilizado en los videos de la gimnasta. Este parámetro se obtiene aplicando el método PCA, el cual es normalmente utilizado para reducción de dimensiones de atributos para compresión de datos. En lo que concierne al uso del método PCA, no se han encontrado artículos acerca de este método para la aplicación de cálculo de orientación de objetos o personas, ni para análisis de trayectorias. El algoritmo aquí presentado, por lo tanto, es una nueva manera de obtener información acerca de la trayectoria de un objeto determinado.

Herramientas para determinación de regularidad de movimiento articulado mediante el seguimiento en una secuencia de imágenes.

5.1.5 Conclusiones sobre las pruebas de iluminación

Aquí nos referimos a las pruebas de iluminación llevadas a cabo con el motor de vaivén tipo péndulo. En cuanto a la variación de iluminación en un mismo cuadro, una iluminación uniforme en la imagen determina una mayor precisión en el seguimiento de las trayectorias. Sin embargo, si la iluminación presenta variaciones pero es aceptable, el umbral podrá mantener la posición del centroide del objeto seguido. Por otro lado, si no se tiene una iluminación constante entre una imagen y las demás, el objeto seguido caerá dentro del umbral establecido para unas imágenes del video pero para otras no será así, ocasionando que se presenten problemas de falsos negativos, o variación en el centroide que está estático. También, hay que tener en cuenta que si ocurre una iluminación diferente entre distintas tomas de video, se tendrá que verificar que tanto esta variación en iluminación desplaza al umbral de color seleccionado en el histograma de la trayectoria de referencia, para saber si los límites de umbral salen de los seleccionados, si es así no se podrá hacer una comparación entre trayectorias. Se recomienda hacer una calibración de la iluminación y el acomodo de la cámara al inicio de una serie de tomas de videos para obtener mejores resultados. Por último, se debe considerar que la cercanía de las lámparas y la profundidad del objeto capturado pueden causar cambios en las sombras del mismo y por lo tanto cambios en la trayectoria percibida por la cámara.

5.1.6 Restricciones, limitantes y contexto de uso

Los algoritmos creados funcionan bajo las siguientes condiciones:

- **Cámara fija:** La cámara con la cual se tomen los videos a analizar debe estar fija y debe tomar la misma posición espacial para la toma de todas los videos.
- **Objetos en movimiento deben estar en todo momento dentro del cuadro de video:** De lo contrario no se podría capturar todo el recorrido del objeto a analizar.
- **Iluminación lo más constante posible para todos los videos, así como para todos los cuadros de cada video:** El algoritmo de segmentación para obtención de trayectorias se basa en detección del color, este tipo de segmentación es susceptible a cambios en la iluminación, lo cual podría causar errores como percepciones de puntos desplazados, falsos positivos y/o falsos negativos.
- **No todos los algoritmos aplican en todo tipo de trayectorias:** La mayoría de los algoritmos están diseñados para trayectorias “abiertas”, sin embargo pueden ser adaptados para trabajar con trayectorias cerradas, mas no todos los parámetros podrán ser adaptables para todas las formas de trayectorias.
- **La definición del objeto detectado no siempre corresponde a su definición física aunque se obtenga un centroide:** Debido a factores como la iluminación y la rapidez de toma de la cámara, los *targets* seguidos podrían no ser percibidos como deberían serlo, modificándose su forma al aplicar los algoritmos de umbralado y segmentación. En la figura 5.1(a) se muestra una imagen con el señalado bien definido, se puede observar que en su correspondiente segmentación la figura 5.1(b) tiene una forma cercana a un círculo. En la figura 5.1(c) se observa que el *target* señalado está difuminado, esto es debido a que la velocidad de captura de la cámara (30 cuadros por segundo) no es lo suficientemente rápida para captar durante el movimiento del motor la forma precisa de los objetos en todos los cuadros. Por ejemplo, esto causa que al llevar a

Herramientas para determinación de regularidad de movimiento articulado mediante el seguimiento en una secuencia de imágenes.

cabo la segmentación, la forma del objeto no se muestra redonda como debería ser, sino que tiene una forma irregular, estrecha en una dirección (figura 5.1(d)), esto es porque solamente el centro del *target* es percibido sin desvanecimiento, mientras que los bordes se difuminan y no caen dentro del rango de umbral establecido. Este problema ocurre en menor medida en el caso de los videos de la gimnasta, puesto que estos videos fueron tomados a una velocidad mayor (120 cuadros por segundo).

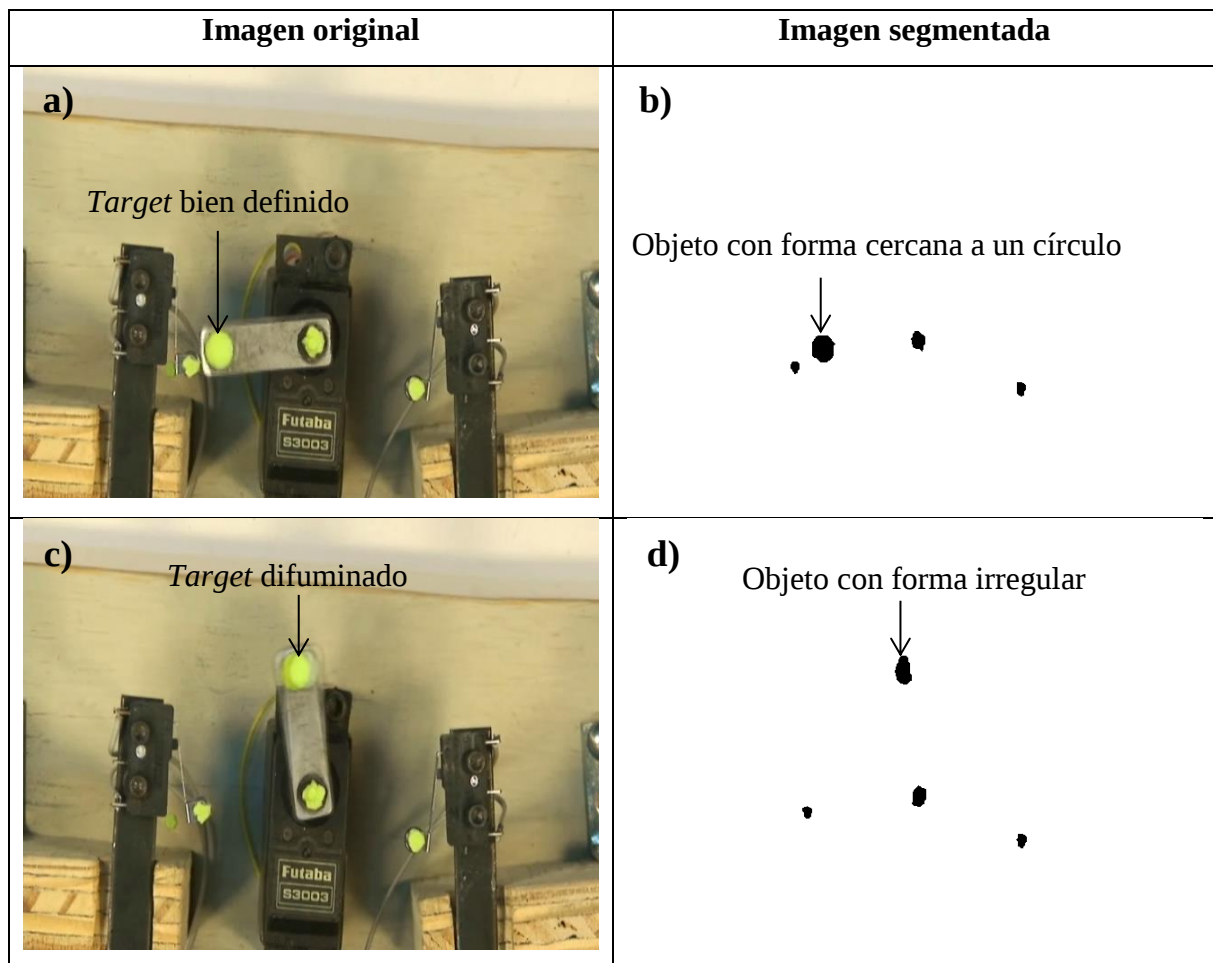


Figura 5.1. Apreciación objetos para distintas posiciones. a) *Target* bien definido, b) Objeto segmentado con forma cercana a un círculo, c) *Target* difuminado, d) Objeto segmentado con forma irregular

Otra cuestión que puede ocasionar una percepción errada de los centroides, es que los *targets* no sean un círculo, esto ocasiona que al realizar la segmentación de las imágenes, los centroides se perciban desplazados del punto que debería ser el centro del *target*, en la figura 5.2 se hace un acercamiento al *target* central de la figura 5.1, para observar este problema. Se puede observar que debido a que el target no es un círculo, se genera una región segmentada diferente al estar en posiciones diferentes, esto aumenta el error de detección, explicado en la sección 4.2.

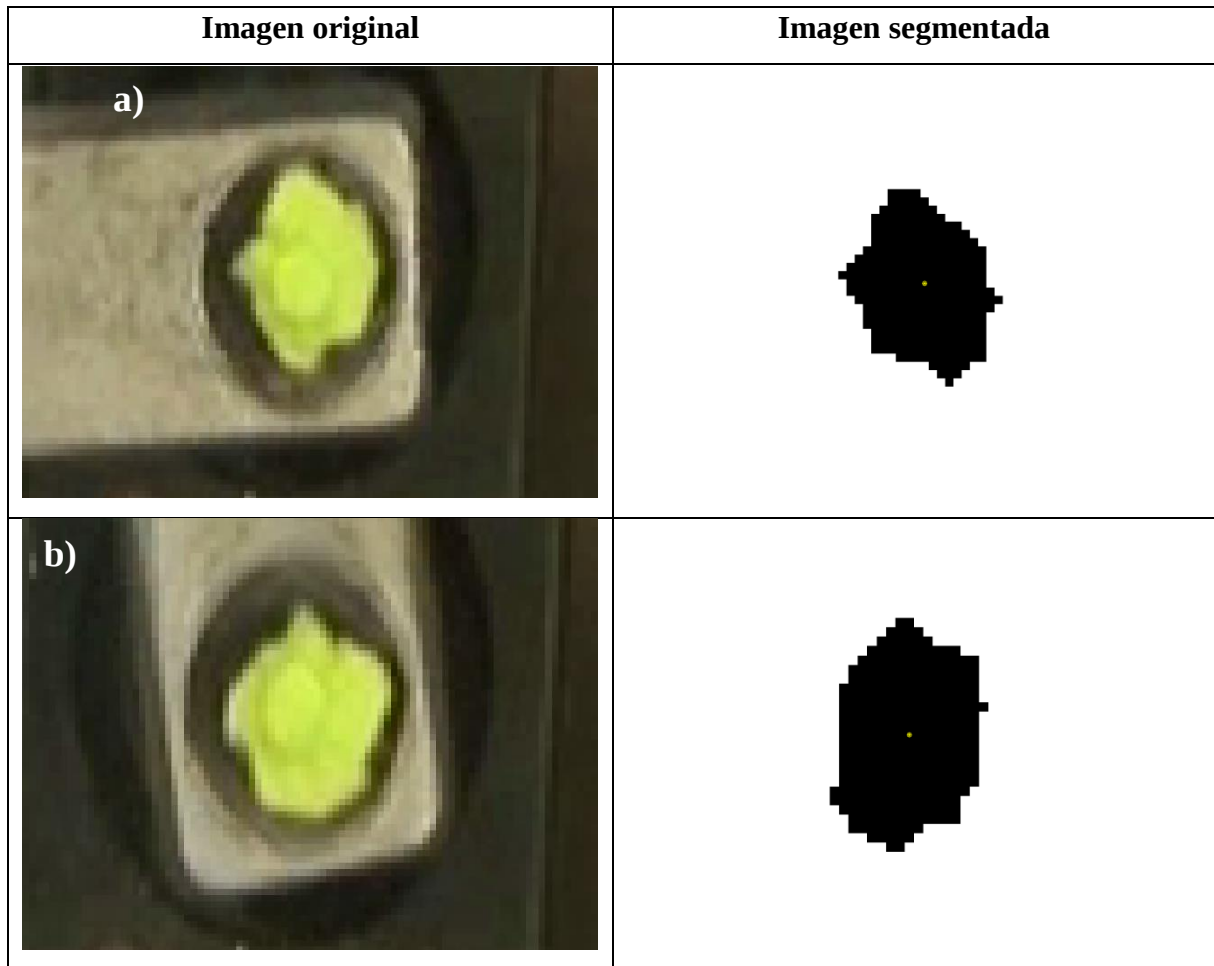


Figura 5.2. Centroides percibidos en posiciones distintas debido a la deformación del target utilizado para detección, a) Posición 1, b) Posición 2.

Herramientas para determinación de regularidad de movimiento articulado mediante el seguimiento en una secuencia de imágenes.

- **Formato no cuadrado del sensor de imagen:** Al observar la figura 3.4 correspondiente a la trayectoria del motor de vaivén, se puede apreciar que esta trayectoria no tiene una forma redonda cuando si debería tenerla, también, si se analizan las medidas de anchura y altura se podrá comprobar que no tiene la misma cantidad de píxeles a lo alto y ancho de la trayectoria, esto es debido a que la percepción de una cámara de video no es necesariamente igual a la de un foto-elemento simétrico, sino que depende de la resolución de la cámara y el formato de pixel sensor de imagen, por lo que es necesario obtener la relación de escala que existe entre la forma real de los objetos y la cámara de video si se desea tener un patrón de referencia dimensional de cuanto equivaldría un pixel en medidas reales.

5.1.7 Posibles aplicaciones

En forma general, se han encontrado hasta el momento tres usos:

- a) Análisis de una trayectoria para determinar parámetros de la misma.
- b) Comparación entre trayectorias para referenciar aspectos de interés de una con respecto a la(s) otra(s).
- c) Utilización de los parámetros de similitud para clasificación de trayectorias.

De estos tres usos generales derivan un sinnúmero de aplicaciones particulares, como por ejemplo:

- Evaluación de trayectorias para medir desempeño de deportistas.
- Medición de patrones en terapias musculares.
- En datos de origen diferente a video: Predicción de trayectorias de huracanes, (ver [16]).
- Monitoreo de mecanismos con movimientos repetitivos.
- Clasificación de trayectorias de tránsito vehicular.

Por supuesto que hay que hacer algunas adaptaciones de los algoritmos y elegir los parámetros de similitud adecuados dependiendo de la aplicación.

5.2 Trabajos futuros

Como trabajos futuros se recomienda hacer lo siguiente:

- Calcular la relación de escala que existe entre la imagen de una cámara de video y el tamaño real de los objetos, para así poder obtener las imágenes en relación con un patrón de referencia dimensional y temporal, tal que se pueda conocer la dimensión que corresponde a una cierta cantidad de píxeles en la imagen, y en qué instante de tiempo.
- Aumentar la cantidad de videos que se tienen como base de datos, puesto que a mayor cantidad de trayectorias, mayor será la posibilidad de aportar datos aceptables como resultados de comparaciones entre estas. Como alternativa, obtener trayectorias de otros objetos.
- Encontrar los objetivos “*targets*” óptimos para establecer los puntos de interés a seguir. Establecer *targets* de forma totalmente redonda para evitar errores de percepción de desplazamiento en los centroides al hacer la segmentación de las imágenes debido a *targets* con forma irregular.
- Aumentar el número de parámetros de similitud y aplicar algoritmos de selección de atributos para elegir los parámetros que aporten mayor información en la distinción de una trayectoria con respecto a otras.

Bibliografía

- [1] S. Lohr, «"The Age of Big Data",» *New York Times* , 2012.
- [2] R. Szeliski, "Computer Vision: Algorithms and Applications", Springer, 2010.
- [3] M. E. G. G. Juan José Cardona Contreras, «"¿QUÉ ALTERACIONES EN VISIÓN BINOCULAR SE PRESENTAN EN PERSONAS CON TRAUMATISMO CRANEOENCEFÁLICO?",» 2002.
- [4] A. O. Assosiation, «American Optometric Assosiation,» 2012. [En línea]. Available: <http://www.aoa.org/x7855.xml>. [Último acceso: Mayo 2012].
- [5] J. F. D. Yiwei Wang, «"Moving Object Tracking in Video",» 2009.
- [6] H. K. J. K. T. a. S. I. Budi Sugandi, «"A Block Matching Technique for Object Tracking Based on Peripheral Increment Sign Correlation Image",» 2008.
- [7] M. J. A. Black, «"A Probabilistic framework for matching temporal trajectories: Condensation-based recognition of gestures and expressions",» 1998..
- [8] M. I. L. Antonis A. Argyros, «"Real-Time Tracking of Multiple Skin-Colored",» vol. LNCS 3023, pp. 368-379, 2005.
- [9] B. B. V. Menser, «"Face Detection and Tracking of Video Coding Applications",» 2000.
- [10] M. R. V. C. D. N. H. Greiffenhagen, «"Statistical Modeling and Performance Characterization of a Real-Time Dual Camera Surveillance System",» 2000.
- [11] T. T. L. G. V. Ferrari, «"Real-Time Affine Region Tracking and Coplanar Grouping",» vol. II, pp. 226-233, 2001.
- [12] T. K. C. T. M. W. W. v. S. U. Handmann, «"Computer Vision for Driver Assistance

- Systems”,» vol. 3364, pp. 136-147, 1998.
- [13] J. D. A. B. S. Intille, «“Real-Time Closed-World Tracking”,» pp. 697-703, 1997.
- [14] A. A. T. D. A. P. C. Wren, « “Pfinder: Analysis and Machine Intelligence”,» vol. 19, pp. 780-785, 1997.
- [15] P. J. A. L. Almeida, M. V. Vieira, M. Kajin y G. F.-M. & R. Cerqueira, «Indices of movement behaviour: conceptual background, effects of scale and location errors,» *ZOOLOGIA*, vol. 27, p. 674–680, 2010.
- [16] S. Dodge, *Exploring Movement Using Similarity Analysis*, Zürich, 2011.
- [17] J. v. V. J. V. A. K. Marco Wiering, «"Intelligent Traffic Light Control",» 2004.
- [18] Microsoft, «Microsoft Kinect,» Microsoft, 2014. [En línea]. Available: <http://www.xbox.com/es-MX/Kinect>. [Último acceso: 20 Marzo 2014].
- [19] P. N. B. G. C. N. G. D. N. F. Broilo Mattia, «Object Trajectory Analysis in Video Indexing and Retrieval Applications,» *Video Search and Mining, Springer*, vol. 287, pp. 3-32, 2010.
- [20] F. R. W. P. Jonas de Vylder, «Segmentation of Cell Nuclei in Arabidopsis Thaliana Roots,» de *Image Analysis and Recognition: 7th International Conference*, Póvoa de Varzim, Portugal, 2010.
- [21] P. E. N. L. Kerstin Witte, «METHOD TO VISUALIZE AND ANALYZE SIMILARITIES OF MOVEMENTS – USING THE EXAMPLE OF KARATE KICKS,» Magdeburg, Alemania, 2010.
- [22] K. O. T. F. Y. O. M. K. M. L. Rieko Osu, «Quantifying the quality of hand movement in stroke patients through three-dimensional curvature,» *Journal of Neuroengineering and rehabilitation*, 2011.

Herramientas para determinación de regularidad de movimiento articulado mediante el seguimiento en una secuencia de imágenes.

- [23] J. M. C. G. Gonzalo Pajares Martinsanz, "Visión por Computador: Imágenes Digitales y Aplicaciones", 2da ed., Madrid: Alfaomega, 2007.
- [24] VICON, «Vicon,» Vicon, [En línea]. Available: <http://www.vicon.com/>. [Último acceso: 20 marzo 2014].
- [25] NDI, «NDI measurement sciences,» [En línea]. Available: <http://www.ndigital.com/msci/products/optotrak-certus/>. [Último acceso: abril 2013].
- [26] S. S. D. K. S. E. K. A. Chino N, «Stroke impairment assessment set,» *Funtional Evaluation of stroke patients*, pp. 19-31, 1995.
- [27] W. R. F. E. Dodge Somayeh, «Revealing the physics of movement: Comparing the similarity of movement characteristics of different types of moving objects,» *Computers, Environment and Urban Systems*, vol. 33, p. 419–434, 2009.
- [28] J. O. S. M. Y. Alper, «"Object Tracking: A Survey",» ACM comput, 2006.
- [29] G. Paschos, «"Perceptually uniform color spaces for color texture analysis: an empirical evaluation",» vol. 10, 2001.
- [30] J. CANNY, «"A computational approach to edge detection",» vol. 8, nº 6, 1986.
- [31] A.-R. M. Amar Mitiche, «"On convergence of the Horn and Shunk optical-flow estimation method",» vol. 13, nº 6, 2004.
- [32] D. Z. C. W. Lei Zhang, «"Robust Object Tracking Using Joint Color-Texture Histogram",» *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 23, nº 7, p. 1245–1263, 2009.
- [33] J. C. Q. Sánchez, «"Estimación de Movimiento de Objetos a Partir de una Secuencia de Imágenes",» 2010.

- [34] C. M. Bishop, *Neural Networks for Pattern Recognition*, Birmingham, UK: Oxford, 1995.
- [35] J. Semmlow, *Signals and Systems for Bioengineers*, San Diego, California: Academic Press, Elsevier, 2011.
- [36] D. F. A. R. Moore L., *Anatomía con orientación clínica*, Barcelona: Williams & Wilkins, 2010.
- [37] A. d. A. d. l. L. Española, "Diccionario de la Lengua Española", Real Academia Española, 2012.
- [38] R. Nathan, «An emergent movement ecology paradigm,» *Proceedings in the National Academy of Sciences*, vol. 105, p. 19050–19051, 2008.
- [39] A. Frank, «Socio-Economic Units: Their Life and Motion,» *Taylor & Francis*, vol. 2, pp. 21-34, 2001.
- [40] S. P. C. D. M. D. M. J. P. F. V. C. Spaccapietra, «A conceptual view of trajectories,» *Data & Knowledge Engineering*, vol. 65, pp. 126-146, 2008.
- [41] W. K. Pratt, "Digital Image Processing", 4 ed., Los Altos, California: PIKS Scientific Inside, 2007.
- [42] J.-B. H. J. P. J. V. P. Gabriel, «"Object Tracking Using Color Interest Points",» 2005.
- [43] L. K. W. J.-H. L. Ssu-Wei Chen, «"Moving Object tracking Based on Background Subtraction Combined Temporal Difference",» 2011.
- [44] M. Piccardi, «"Background subtraction techniques: a review",» 2004.
- [45] J. M. Jianbo Shi, «"Normalized Cuts and image segmentation",» vol. 22, nº 8, 2008.
- [46] H. Hamedivafa, «"Application of an Improved Method of Image Segmentation and Some Considerations on Identification and Tracking of Umbral Dots",» 2008.

Herramientas para determinación de regularidad de movimiento articulado mediante el seguimiento en una secuencia de imágenes.

- [47] R. B. A. J. M. K. E. Atkociunas, «"Image Processing in Road Traffic Analysis",» 2005.
- [48] S. K. Kardarla, «"Object Tracking Video Images Based on Image Segmentation and Pattern Matching",» 2009.
- [49] R. Rojas, Neural Networks A Systematic Introduction, Berlín: Springer, 1996.
- [50] D. & M. P. Comancianu, «"Mean Shift: A robust approach toward feature space analysis",» vol. 24, nº 5, 2002.
- [51] A. W. Michael Kass, «"Snakes: Active contour models",» 1988.
- [52] R. W. R. Gonzales, "Digital Image Processing", 2da ed., Reading, Massachusetts: Addison-Wesley, 1993.
- [53] A. Rosenfeld, «"Survey: Analysis and Computer Vision: 1999",» vol. 78, pp. 222-302, 2000.
- [54] B. S. Mehmet Sezgin, «"Survey over image thresholding techniques and quantitative performance evaluation",» 2004.
- [55] R. M. K. Dan George Bucatanschi, «"Kernel Methods for Image Processing",» 2006.
- [56] A. P. T.S. Jebara, «"Parameterized structure from motion for 3d adaptive",» 1997.
- [57] N. K. S. A. H. K. S.H. Kim, «"Object oriented face detection using range and color information.",» pp. 76-81, 1998.
- [58] R. F. D. Saxe, «"Toward robust skin identification in video images.",» pp. 379-384, 1996.
- [59] K. N. D. Chai, «"Locating facial region of a head-and-shoulders color image.",» pp.

124-129, 1998.

[60] N. A. M.H. Yang, «"Detecting human faces in color images."», pp. 127-130, 1998.

[61] f. S. K. H. H. R. V. B. F. W. S. J. H. N. Rohrer B, «Movement smoothness changes during stroke recovery,» *The journal of neuroscience* , pp. 8297-8304, 2002.

[62] G. C. M. a. J. Gan, «ASA-SIAM Series on Statistics and Applied Probability,» de *Data Clustering: Theory, Algorithms, and Applications*, Philadelphia, SIAM, 2007, pp. 71-75.

[63] A.-R. M. [13] Amar Mitiche, «"On convergence of the Horn and Shunk optical-flow estimation method",» vol. 13, n° 6, 2004.

APÉNDICE A

Glosario de términos y conceptos

A continuación se presenta un glosario de términos utilizados a lo largo de la presente tesis, algunos de estos conceptos son realmente básicos pero es necesario tenerlos en cuenta para una mayor comprensión de la investigación, la mayoría de las definiciones fueron extraídas del diccionario de la Real Academia Española [37].

A.1. Conceptos generales

- **Pixel (Picture Element):** Superficie homogénea más pequeña de las que componen una imagen digital, que se define por su brillo y color, y que tiene ubicación dentro de un sistema de coordenadas (x,y) de dicha imagen.
- **Imagen:** Una imagen es la reproducción de la figura de un objeto por la combinación de los rayos de luz que proceden de él [37].
- **RGB (Red, Green, Blue):** Colores primarios desde el punto de vista informático, los cuales se subdividen para formar los demás colores, representan una imagen a colores por medio de tres matrices.
- **Saturación:** Indicación de la cantidad de color, dependiente de la resolución.
- **Luminancia:** Cantidad de luz o brillo en una imagen.
- **Umbral:** Valor mínimo de una magnitud a partir del cual se produce un efecto determinado. Hablando en términos de imágenes, el umbral es el valor de intensidad a partir del cual se establece un cambio o se selecciona una región de píxeles, tanto para una imagen en escala de grises como para una imagen RGB.

- **Luxómetro:** Instrumento de medición de la iluminancia de un ambiente.
- **Lux:** Unidad derivada del Sistema Internacional de Unidades para la medición de la iluminancia o nivel de iluminación, equivale a un lumen/m^2 .
- **Lumen (lm):** Es la unidad del Sistema Internacional de Unidades para la medición del flujo luminoso, una medida de la potencia luminosa emitida por una fuente.
- **Centroide:** Punto que define el centro geométrico de un objeto.
- **Dilatación:** Operación morfológica aplicada a imágenes digitales binarias que consiste en la adición de píxeles alrededor de una zona de píxeles con valor de intensidad de 1.
- **Erosión:** Operación morfológica aplicada a imágenes digitales binarias que consiste en la eliminación de píxeles alrededor de una zona de píxeles con valor de intensidad de 1.
- **Cerradura:** Operación morfológica consistente de una operación de dilatación seguida de una operación de erosión, normalmente utilizada para definir mejor ciertas zonas de una imagen binaria.
- **Movimiento:** Es el cambio de ubicación espacial de un individuo u objeto en el tiempo [38]. En [39] se definen dos tipos de cambio en general, “Cambio en los objetos de interés” (aparición, desaparición, unión o división), así como “Cambios en la posición o forma de los objetos de interés” (movimiento), estos últimos son los que se toman en cuenta en la presente tesis.
- **Trayectoria:** Una trayectoria se puede definir como la representación espacio-temporal del movimiento de un individuo [40], para razones prácticas de mediciones y observaciones de la posición de un objeto en movimiento con respecto al tiempo, una trayectoria consiste en una serie de observaciones discretas espaciadas en el tiempo, en la figura a.1 se aprecia un ejemplo de una trayectoria seguida por una hormiga, cada punto de la trayectoria representa una posición en la cual la hormiga estuvo en un

Herramientas para determinación de regularidad de movimiento articulado mediante el seguimiento en una secuencia de imágenes.

determinado momento, desde la posición f_1 hasta f_n donde n es la cantidad total de posiciones de la trayectoria, cada punto de la trayectoria contiene tres datos, $f(X,Y,t)$ donde X e Y son las coordenadas espaciales de ese punto, mientras que t es el tiempo en el cual la hormiga se encontraba en esa posición.

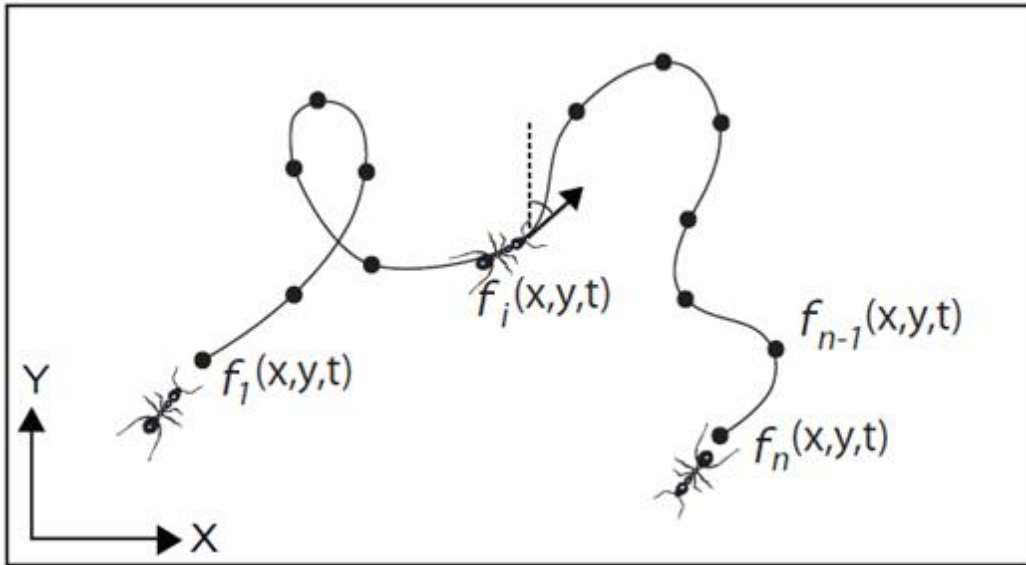


Figura A.1. Trayectoria de un objeto en movimiento: Representación espacio-temporal del movimiento de un objeto (Imagen obtenida de S. Dodge, “Exploring Movement Using Similarity Analysis”, [16]).

- **Parámetros de movimiento (similitud):** Consisten en una serie de características que pueden ser utilizadas como variables de comparación o caracterización de movimientos.
- **Análisis de Componentes Principales:** Este algoritmo, como su nombre lo indica, adquiere los componentes principales de un conjunto de datos de acuerdo a las direcciones en las cuales se tiene una mayor varianza, y deja a un lado aquellos datos que no son relevantes. El algoritmo de PCA es comúnmente utilizado para reducción

dimensión en atributos de clasificación y compresión de datos, en lo que a este trabajo respecta, se tiene una implementación similar, pero un fin distinto.

- **MATLAB:** Es un lenguaje de alto nivel y un entorno interactivo para el cálculo numérico, la visualización y la programación.

A.2. Histograma y sus propiedades estadísticas

- **Histograma:** Representación gráfica de una distribución de frecuencias por medio de rectángulos, cuyas anchuras representan intervalos de la clasificación y cuyas alturas representan las correspondientes frecuencias. El histograma de una imagen es una función discreta que representa el número de píxeles de una imagen en función de los niveles de intensidad (ver glosario de términos para una definición general de histograma). En el caso de una imagen en niveles de gris, la probabilidad de ocurrencia de un determinado nivel de gris se puede representar como $P(g)$, mientras en el caso de una imagen RGB se tendría un histograma con tres tipos de probabilidades $P(R)$, $P(G)$, $P(B)$ (ver figura a.2). Las probabilidades de ocurrencia de cada color se definen como [23].

$$P(R) = \frac{N(R)}{M}$$

$$P(G) = \frac{N(G)}{M}$$

$$P(B) = \frac{N(B)}{M}$$

Donde M es número de píxeles de la imagen y N es el número de píxeles en el nivel de la intensidad del color correspondiente.

Herramientas para determinación de regularidad de movimiento articulado mediante el seguimiento en una secuencia de imágenes.

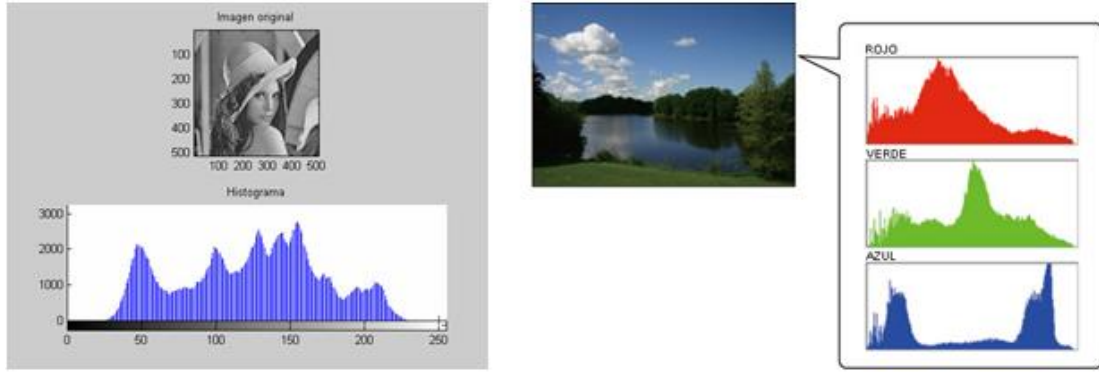


Figura A.2. Histograma representativo de una imagen en escala de grises (izquierda) y una imagen a color RGB (derecha).

Cada pixel se puede codificar con un byte (8 bits) por lo que admite hasta 256 variaciones de color (2^8 posibilidades binarias) como se puede observar en el histograma en escala de grises de la figura a.2; lo mismo ocurre para las imágenes RGB, pero esta vez el nivel de intensidad es por color. A mayor intensidad en los píxeles de una imagen, el histograma estará más “cargado” hacia la parte derecha (los píxeles tendrán valores más elevados) y viceversa.

A continuación se mencionan algunas propiedades estadísticas del histograma que informan sobre la distribución de intensidades de cada uno de los colores RGB o, en su caso, de los niveles de gris de una imagen. Las propiedades y ecuaciones se describirán en términos de imágenes de niveles de gris, pero pueden ser aplicadas para cada uno de los colores primarios en una imagen RGB [41].

- **Media:** Es el valor medio de los niveles de gris o de un determinado color RGB, está definida por la ecuación (A.1):

$$\bar{g} = \sum_{g=0}^{L-1} gP(g) = \sum_x \sum_y \frac{f(x,y)}{M} \quad (\text{A.1})$$

Donde L es el número total de niveles de gris (o color).

- **Varianza:** Mide la dispersión a los alrededores de la media, está definida por la ecuación (A.2):

$$\sigma^2 = \sum_{g=0}^{L-1} (g - \bar{g})^2 P(g) \quad (\text{A.2})$$

- **Desviación estándar:** Es una medida de centralización y se define como la raíz cuadrada de la varianza y está dada por la ecuación (A.3).

$$\sigma = \sqrt{\sum_{g=0}^{L-1} (g - \bar{g})^2 P(g)} \quad (\text{A.3})$$

Herramientas para determinación de regularidad de movimiento articulado mediante el seguimiento en una secuencia de imágenes.

A.3. Mecanismos de detección de objetos

Todos los métodos utilizados en seguimiento de objetos requieren un mecanismo de detección. Al igual que para elegir las características de seguimiento (color, bordes, flujo óptico y texturas), existen diversos métodos para realizar la detección de objetos, el método que se seleccione también dependerá de las aplicaciones requeridas y del criterio de quien las implemente. A continuación se enlistan los métodos más populares [28].

A.3.1 Detectores puntuales

Los detectores puntuales son utilizados para encontrar puntos de interés en imágenes que tienen diversidad de texturas. Una cualidad deseable de éste método es su invariabilidad a cambios de iluminación y punto de vista de la cámara (Ver figura A.3).

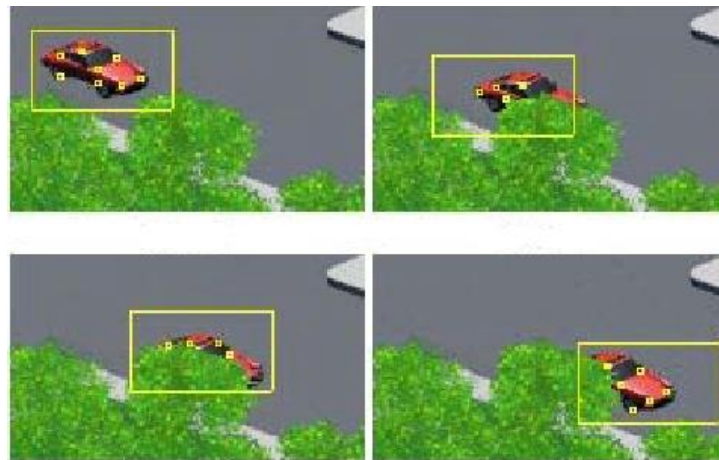


Figura A.3 Ejemplo de detección de objetos mediante detectores puntuales (Imagen tomada de P. Gabriel, "Object Tracking Using Color Interest Points", [42]).

A.3.2 Substracción de fondo

La detección de objetos puede ser realizada al construir una representación de una escena llamada modelo de fondo, para después encontrar desviaciones de este modelo para cada cuadro posterior. Cualquier cambio significativo en alguna región de una imagen a partir del modelo de fondo significa que un objeto se ha movido. Los píxeles que constituyen una región sometida a cambios son marcados para procesamiento posterior. Usualmente, se aplica un algoritmo de conexión para obtener regiones conectadas correspondientes a los objetos. En la figura A.4 se presenta un ejemplo de este método.



Figura A.4. Ejemplo de método de substracción de fondo para detección de objetos (Imagen tomada de L. Ssu-Wei Chen, "Moving Object tracking Based on Background Subtraction Combined Temporal Difference", [43]).

Entre algunas de las desventajas de este método se encuentra el hecho de que requiere de una gran cantidad de memoria en relación con los demás, además de que la precisión en la diferenciación de cuadros se va perdiendo como consecuencia del incremento en la velocidad de los objetos a seguir y la tasa de cambio de cuadro a cuadro [44].

A.3.3 Segmentación

El objetivo de los algoritmos de segmentación de imágenes es la partición de una imagen en regiones perceptualmente similares. El atributo de segmentación más básico es el de luminiscencia para una imagen monocromática (escala de grises) o los componentes

Herramientas para determinación de regularidad de movimiento articulado mediante el seguimiento en una secuencia de imágenes.

de color en una imagen a color [41]. Todo algoritmo de segmentación aborda dos problemas, el criterio para una buena partición y el método para alcanzar una división eficiente [45]. La figura A.5 muestra un ejemplo del método de segmentación; en esta figura puede observarse que el método de segmentación está basado en el tratamiento del histograma (b), (d) y (f), además de la selección de un umbral para determinar las zonas a segmentar, este método tiene la ventaja de que es muy simple de implementar una vez encontrado el valor de umbral para la segmentación de la imagen, por otra parte, hay ocasiones en que encontrar este valor es sumamente difícil, aun así, este método ha resultado ser uno de los más usuales debido a que ha brindado excelentes resultados en otros trabajos [33] [46] [47] [48].

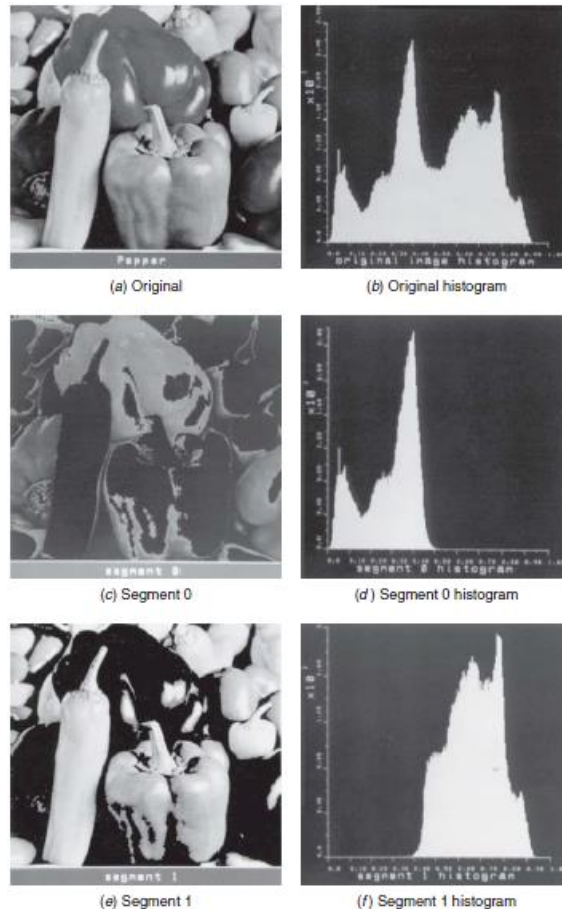


Figura A.5. Ejemplo del método de segmentación para detección de objetos (Imagen tomada de “W. K. Pratt, Digital image processing”, [41]).

A.4. Definición de red neuronal

La neurona es la unidad básica que compone el cerebro humano, a partir de este tipo de células se generan los procesos de razonamiento. Toda neurona se constituye de tres partes principales, las cuales son soma, axón y dendritas. El proceso de comunicación entre dos o más neuronas se llama sinapsis y se lleva a cabo por medio de las interconexiones de sus dendritas (reciben información) conectadas a su vez a los axones (envían información), mientras que el soma se encarga del procesamiento de la información [34].

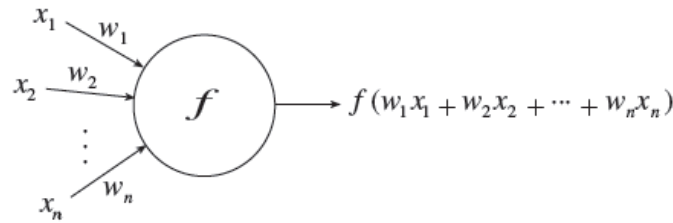


Figura A.6. Representación de una neurona en forma artificial [49].

En analogía con las neuronas naturales, existen sistemas y/o algoritmos que funcionan a modo de neuronas artificiales, estas se comunican entre sí mediante conectores que se encargan de llevar la información desde y hacia las demás neuronas, la figura a.6 muestra la estructura de una neurona con n -entradas. Cada entrada i se representa como X_i . Para cada neurona existe una función primitiva f , la cual se encarga de evaluar las entradas para una serie de pesos w asociados a cada entrada, es decir, cada entrada X_i se multiplica por un peso w_i y se evalúa mediante la función f (analogía del soma) para encontrar una salida definida. La figura a.6 es un ejemplo de una sola neurona, sin embargo, como en la naturaleza, se necesita de una agrupación de neuronas para tener un resultado eficiente, a esta agrupación en su forma artificial se le conoce como red neuronal. Una red neuronal es un sistema que trata de emular el comportamiento de las neuronas del cerebro humano para resolver problemas de aprendizaje de máquinas, por ejemplo, clasificación de datos o patrones de datos y ajuste de curvas a un modelo de datos establecido [49]. La figura a.7 muestra la representación esquemática de una red neuronal.

Herramientas para determinación de regularidad de movimiento articulado mediante el seguimiento en una secuencia de imágenes.

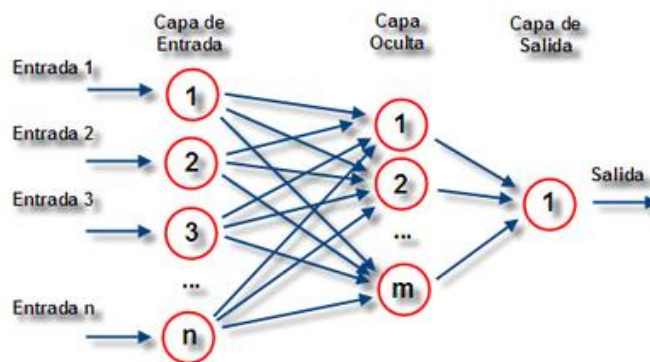


Figura A.7. Representación esquemática de una red neuronal.

Las redes neuronales están constituidas por al menos dos capas de neuronas: una capa de entrada y una capa de salida; sin embargo en aplicaciones complejas también se requiere de la agregación de una o más capas internas. Cada función interna f de cada neurona que conforma la red contribuirá a la obtención de resultados al actualizar los pesos w . Las redes neuronales artificiales pueden ser utilizadas tanto para problemas de clasificación como problemas de regresión. Los problemas de clasificación se refieren a la agrupación de datos dependiendo de patrones similares encontrados en los mismos, por ejemplo, la determinación de si un tumor es maligno o benigno (1 o 0) dependiendo de su forma, tamaño, etc.; a su vez, la regresión aborda los problemas de ajuste de datos a un modelo definido, por ejemplo, determinación del precio de una casa dependiendo del área del terreno, número de cuartos, etc.; por ende, se puede concluir que los problemas de regresión trabajan con resultados continuos, mientras que en los problemas de clasificación los resultados son discretos.

APÉNDICE B

Algoritmos diseñados

A lo largo de este apéndice se presentan todos los algoritmos creados durante la presente investigación, los algoritmos están comentados en español para un mejor entendimiento de lo que se hace en cada paso. Los Algoritmos se presentan en el orden que deberían ser utilizados.

B.1. Algoritmo para generación de trayectorias (edp_Al).

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                               Descripción:
% Extrae puntos de un conjunto de imágenes de una trayectoria y crea una
% tabla (X,Y) de las coordenadas de los puntos extraídos de dicha
% trayectoria.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Carga video a analizar

video = VideoReader(Video)

N = video.NumberOfFrames; %Total de cuadros del video

Rate = video.FrameRate;
Duracion = video.Duration;

vidAlto = video.Height;
vidAncho = video.Width;

%Creación de un fondo blanco ("lienzo" donde se graficará la
trayectoria).
W = ones([vidAlto, vidAncho]);

indices = [];
cxAnt = [];
cyAnt = [];

%Lectura de las imágenes desde 1 hasta el total de imágenes
for i = 285:N

    %Lee imagen en turno
    imagen = read(video,i);

```

Herramientas para determinación de regularidad de movimiento articulado mediante el seguimiento en una secuencia de imágenes.

```
%Obtiene los centroides de los objetos con el uso de la función
Salto.
C = salto(imagen);

%Pregunta si la matriz de centroides C no está vacía, es decir, si
se ha detectado al menos una centroeide en la imagen en turno
if (~isempty(C))

    %Redondea los centroides al pixel más cercano.
    C = round (C);

    %Obtiene el numero de filas de la matriz de centorides para
    graficarla
    B = size(C,1);

    %Desde 1 hasta el numero de renglones de la matriz de
    centroides.
    for j = 1:B

        %El pixel correspondiente al centroeide de la fila en
        turno sera negro dentro del fondo blanco creado.
        W(C(j,2),C(j,1)) = 0;
        cxAnt = [cxAnt ; C(j,1)];
        cyAnt = [cyAnt ; C(j,2)];
        indices = [indices ; i];

    end
end

%Mostrar imagen resultante.
imshow(W);

%Se pasan datos a vectores de puntos.
X = cxAnt;
Y = cyAnt;

%Y se "voltea" porque en una imagen las coordenadas estan al revés de las
%coordenadas de un sistema (X,Y) convencional.
Y = size(W,1) - Y;

%Se guardan los datos en una matriz llamada Puntos
Puntos = [X Y indices];
```

B.2. Función de asignación de umbral para videos de salto

```
function[centroids,numObjects] = salto_A1(I)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                               Descripción:
%
% Obtiene el centroide de una gimnasta en una secuencia de imágenes
% mediante detección de color dado un umbral.
%
% Entradas:
% * I: Imagen de entrada.
%
% Salidas:
% * centroids: Coordenadas del centroide detectado.
% * numObjects: número de objetos detectados.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

imR=double(I(:,:,1));
imG=double(I(:,:,2));
imB=double(I(:,:,3));
[N,M]=size(imG);
Y=zeros(N,M);
for i=1:N;
    for j=1:M;
        if((imR(i,j)>83.44 && imR(i,j)<109.78) && (imG(i,j)>25.89 &&
imG(i,j)<46.05) && (imB(i,j)>43.02 && imB(i,j)<65.1))
            Y(i,j)=1;
        else
            Y(i,j)=0;
        end
    end
end

%Aplica función de cerradura
se=strel('disk',20);
CY=imclose(Y,se);
bw=bwareaopen(CY,20); %25

% se=strel('square',10);
% bw=imdilate(bw,se);
bw=imfill(bw,'holes');

%Extrae centroides de la imagen resultante
L=logical(bw);
s=regionprops(L,'Area','centroid');
centroids=cat(1,s.Centroid);

[labeled,numObjects]=bwlabel(bw,8);
end
```

Herramientas para determinación de regularidad de movimiento articulado mediante el seguimiento en una secuencia de imágenes.

B.3. Función de selección de puntos para eliminar falsos positivos

```
function [TL,indices] = selectPoints_Al(Puntos)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                               Descripción:
% Esta función "limpia" una trayectoria al permitir seleccionar sólo los
% puntos que sean de interés en esta trayectoria, dejando fuera aquellos
% puntos que no correspondan al objeto u objetos seguidos.
%
% Entradas:
% * Puntos: Coordenadas de la trayectoria original.
% Salidas:
% * TL: trayectoria con puntos de interés seleccionados.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Separación de la matriz puntos en coordenadas (X,Y).
X = Puntos(:,1);
Y = Puntos(:,2);
indices = Puntos(:,3);
a = cat(3,X,Y,indices);

%Se grafica la trayectoria sin "limpiar".
plot3(X,Y,indices,'ob');
view(2);
grid on;
%Se activa la brocha para selección.
brush on

%Imprime un mensaje y espera hasta que se oprima una tecla.
fprintf('\nSeleccione los puntos de interés con la herramienta brocha,\n\noprima click derecho-copy data to clipboard y presione cualquier tecla para continuar:\n\n')
pause;

%Los puntos seleccionados se guardan en la matriz Puntos.
Puntos = clipboard('paste');

%Se convierte la matriz Puntos de variables tipo string a números.
trayectoriaLimpia = str2num(Puntos); %#ok<*ST2NM>

TL = trayectoriaLimpia(:,1:2);
indices = trayectoriaLimpia(:,end);
%Se grafica la trayectoria limpia
plot(TL(:,1),TL(:,2),'*r');
grid on

end
```

B.4. Algoritmo de entrenamiento de red neuronal para ajuste de curvas

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%                               Descripción:
% Esta función entrena una red neuronal para un conjunto de datos
% realizando los siguientes pasos:
%
% - Establece parámetros de entrenamiento: Capas internas, función de
%   aprendizaje, estrategia de aprendizaje, parámetro de evaluación
% - Divide datos en entrenamiento (70%), validación (15%) y prueba(15%)
% - Entrena la red bajo los parámetros configurados
% - Grafica el ajuste de los datos
% - Predice los datos de test y los grafica junto con los datos reales
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Carga los datos.
data1 = load('AntePenultimoSalto.mat');

X1 = data1.X;
Y1 = data1.Y;

%Datos de entrenamiento transpuestos para que funcionen con las variables
%de MATLAB
P = X1';
T = Y1';

%datos que se usan para la simulación
FP=X1';
FT=Y1';

% Unidades y capas internas
LY=20;
TF={'tansig','purelin'};

% Estrategia de aprendizaje a utilizar.
% Opciones: trainlm, trainbfg, trainrp, traingd,trainscg
BTF='trainlm';

%Función de aprendizaje gradient descent
BLF='learngd';

>Error cuadrático medio como valor de evaluación.
PF='mse';

net1 = newff(P,T,LY,TF,BTF,BLF,PF); %Crea la Red

net1.plotFcns={'plotperform','plotregression'}; %Grafica de regresión

net1.divideParam.trainRatio = 70/100; %Entrenamiento = 70%
net1.divideParam.valRatio = 15/100; %Validación=15%
net1.divideParam.testRatio = 15/100; %Prueba = 15%

```

Herramientas para determinación de regularidad de movimiento articulado mediante el seguimiento en una secuencia de imágenes.

```
net1.trainparam.epochs = 1000;
net1.trainParam.goal = 0.01;           %Meta para el error
net1.trainParam.lr = 0.01;            %Tasa de aprendizaje

%Se entrena la red
[net1,tr,Y,E]=train(net1,P,T);

%Generación de variables
A = 847;
var = [A ; A^2 ; A^3];

[Y,Pf,Af,E,perf] = sim(net1,var);

YT = sim(net1,FP);
%Grafica de ajuste de los datos obtenidos a los datos reales
plotregression(FT,YT);
%Traspone para hacer comparación y graficar
YT=YT';
FT=FT';

datos = horzcat(FT,YT);

figure;

%Configuración de ejes
axes1 = axes('Parent',figure1);
box(axes1,'on');
hold(axes1,'all');

%Se grafica matriz con las dos series de valor real y predicción
plot1 = plot(10*(1:size(FT,1)),datos,'Parent',axes1,'LineWidth',2);

%Apariencia de serie de valor real y predicción
set(plot1(1),'MarkerFaceColor',[0 0 1],'Marker','square','Color',[0 0
1],'DisplayName','Predicción');
set(plot1(2),'MarkerFaceColor',[1 0 0],'Marker','o','Color',[1 0
0],'DisplayName','Valor real');

%Crea leyenda para gráfica
legend(axes1,'show');
```

B.5. Función de interpolación mediante mínimos cuadrados

```
function [Ec_Salida, p, ECM_propio, mu, sigma] =
interpolación_Al(datos_Entrada,orden,varargin)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%                               Descripción:
% Este programa ejecuta la interpolación y calcula una función polinómica
% para un conjunto de datos (X,Y).
%
% Entradas:
%
% * datos_Entrada: Datos de entrada, donde la primera columna corresponde
%   a las coordenadas de la trayectoria en X y la segunda columna a las
%   coordenadas en Y.
% * orden: Orden del polinomio de salida (Escalar).
% * varargin: si es 1, plotea la gráfica con la función encimada.
%
% Salidas:
%
% * Ec_Salida: Representación del polinomio de interpolación mediante una
%   ecuación en función de x.
% * p: Polinomio de interpolación de los datos (X,Y).
% * ECM_propio: Error cuadrático medio del polinomio de interpolación al
%   ser evaluado para los mismos datos con los que se creó
%   (error de interpolación).
% * mu: media de todos los datos, se necesita para cuando se evaluarán
%   puntos nuevos, estos se deben normalizar con mu y sigma.
% * sigma: desviación estándar de los datos, se necesita para cuando se
%   evaluarán puntos nuevos, estos se deben normalizar con mu y sigma.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Creación de variables (X,Y) por separado
X = datos_Entrada(:,1);
Y = datos_Entrada(:,2);

%[X_Uni Y_Uni] = prom_Ys(datos_Entrada); %Funcion de pormediacion
%de Y's
%para X iguales.

%Normalización de los datos para ejecutar la interpolación
%X_normalizada = X-media/desviación estándar
mu = mean(X);
sigma = std(X);
X_Norm = ((X)-repmat(mu,size(X,1),1))./sigma;

% Retiene gráfica para graficar la interpolación junto con los
puntos de centroides.
hold on

% polyfit obtiene un polinomio "p" para una serie de datos (X,Y,N),
donde N será el grado del polinomio.
[p] = polyfit(X_Norm, Y, orden);
```

Herramientas para determinación de regularidad de movimiento articulado mediante el seguimiento en una secuencia de imágenes.

```
% xc es el valor o serie de valores en los cuales el polinomio de
interpolación se va a evaluar.
xc = min(X_Norm) : 0.01 : max(X_Norm);

%Y_Ev será la evaluación del polinomio de interpolación "p", para
obtenerlo se ejecuta la función polyval la cual requiere como
argumentos el polinomio a evaluar "p" y los valores de X para los
cuales se quiere evaluar.
Y_Ev = polyval(p, xc);

%Se convierte X a su valor original (antes de haber sido
normalizado), para que los valores de Y correspondan al valor
original de X en la gráfica %esto es solo para visualización.
X_Ev = (xc*sigma) + mu;

%Se grafica el resultado para visualización
if cell2mat(varargin) == 1;
    plot(X_Ev, Y_Ev, '-k', 'linewidth', 1)
    grid on
end

% Se crea una variable literal x para insertar en una ecuación de
salida
% representativa del polinomio de interpolación
syms x

% Se crea un vector de tamaño del polinomio de interpolación con x
con
% exponente desde 0 hasta el grado del polinomio.
v = x.^(0:length(p)-1);
v = fliplr(v);

% La ecuación de salida representativa se obtiene al multiplicar el
% vector de x's con el vector polinomio.
Ec_Salida = sum(expand(v.*p));

% Evaluación de los puntos para el polinomio obtenido con respecto
a los
% mismos puntos con los que se obtuvo para obtener un error
cuadrático %medio de sí mismo (error de interpolación)
YY = polyval(p, (X-mu)/sigma);
%Error cuadrático medio
ECM_propio = (sum((Y - YY).^2))/length(YY);

%NOTA IMPORTANTE: Si se desea evaluar un valor nuevo de X en la
función polyval, este valor debe ser normalizado primero, con la
misma mu y sigma que se calcularon para la extracción del polinomio
de interpolación. EJEMPLO: si se desea evaluar el valor X=150, se
tendría que hacer la siguiente operación:
%xc = (X-mu)/sigma
%Y = polyval(p,xc)
%La Y obtenida será la correspondiente al valor X=150;

end
```

B.6. Función de evaluación de función de interpolación

```
function [Y_Ev] = evPuntos_A1(A,p,mu,sigma,rango)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                               Descripción:
%
% Esta función evalúa un conjunto de puntos para una función polinomial
% Los puntos son normalizados antes de ser evaluados (para eso es la
% media(mu) y la desviación estándar (sigma)).
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Crea un rango de valores a evaluar
xc = min(A(:,1)):rango:max(A(:,1));

%Normaliza el valor para que pueda ser evaluado
xc = (xc-mu)/sigma;

%Evaluación de los valores de X para el polinomio p dado
Y_Ev = polyval(p,xc);
Y_Ev = Y_Ev';

end
```

Herramientas para determinación de regularidad de movimiento articulado mediante el seguimiento en una secuencia de imágenes.

B.7. Función de obtención de error cuadrático medio

```
function [ECM] = ecmT_A1(datos_Entrada,p,mu,sigma)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                               Descripción:
% Esta función evalúa o compara un conjunto de puntos con respecto a una
% función de interpolación dada, obteniendo el error cuadrático medio ECM
% entre ambos, es decir, la diferencia entre los valores Y de los puntos
% a evaluar y los valores Y determinados por la función interpolación.
% Esto se hace para todos los valores de X contenidos en los datos a
% evaluar.
%
% Entradas:
%
% X: Vector de coordenadas en X de los datos a evaluar.
% Y: Vector de coordenadas en Y de los datos a evaluar.
% p: polinomio de interpolación del cual se obtiene la función con
% respecto a la que se lleva a cabo la evaluación.
% mu: media de todos los datos con los cuales se obtuvo el polinomio de
% interpolación.
% sigma: Desviación estándar de todos los datos con los cuales se obtuvo
% el polinomio de interpolación.
%
% Salidas:
%
% ECM: Error cuadrático medio resultante de la evaluación para determinar
% que tan diferentes son los puntos evaluados con respecto a la función
% interpolada.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

X = datos_Entrada(:,1);
Y = datos_Entrada(:,2);

%Se normalizan las variables para su evaluación.
xc = (X-mu)/sigma;

%Evaluación de los valores de X para el polinomio p dado
Y_Ev = polyval(p,xc);

%Obtención del Error Cuadrático Medio
ECM = (sum((Y - Y_Ev).^2))/length(Y_Ev);
end
```

B.8. Función de correlación y covarianza cruzadas

```
function [corr,cov] = xcorr cov_A1(A,B)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                               Descripción:
%
% Esta función obtiene la correlación cruzada y covarianza cruzada para
% dos conjuntos de coordenadas (X,Y) correspondientes a una trayectoria,
% aun cuando los conjuntos de puntos no tengan el mismo tamaño.
%
% Entradas:
%
% * A: Conjunto de coordenadas o puntos 1.
% * B: Conjunto de coordenadas o puntos 2.
%
% Salidas:
%
% * C: Correlación entre A y B.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Número de datos en A y B.
nA = size(A,1);
nB = size(B,1);
nVar = size(A,2);

% Diferencia del el número de datos entre conjunto A y B.
dif_Datos = abs(nA-nB);

% Agrega ceros al conjunto más pequeño hasta que sea del tamaño del
% más grande.
if nA >= nB
    B = [B ; zeros(dif_Datos,nVar)];
else
    A = [A ; zeros(dif_Datos,nVar)];
end

% Obtiene correlación cruzada y covarianza cruzada para los conjuntos
% de datos.

corr = xcorr(A,B,'coeff');
cov = xcov(A-mean(A),B-mean(B),'coeff');

end
```

Herramientas para determinación de regularidad de movimiento articulado mediante el seguimiento en una secuencia de imágenes.

B.9. Función de obtención de puntos críticos

```
function [minT,maxT] =  
pCriticos_Al(datos_Entrada,Ec_Interpolada,mu,sigma)  
  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
%                               Descripción  
% Esta función calcula los puntos críticos de una función matemática  
% (Máximos y mínimos) dados los coeficientes de la función.  
%  
% Entradas:  
%  
% * datos_Entrada: Datos de entrada, donde la primera columna corresponde  
%   a las coordenadas de la trayectoria en X y la segunda columna a las  
%   coordenadas en Y.  
% * Ec_Interpolada: Vector de los coeficientes de la función creada para  
%   los datos de entrada.  
% * mu: media de los datos de entrada.  
% * sigma: desviación estándar de los datos de entrada.  
%  
% Salidas:  
%  
% * minT: Matriz de coordenadas de mínimos.  
% * maxT: Matriz de coordenadas de máximos.  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
  
% Obtención de primera y segunda derivada  
pd = diff(Ec_Interpolada,1);  
sd = diff(Ec_Interpolada,2);  
  
% Obtiene coeficientes de la primera derivada, se debe poner como  
% [c ~]  
%para que obtenga los coeficientes en el orden que debe ser.  
[c,~] = coeffs(pd);  
  
% Convierte los coeficientes de variables simbólicas a variables  
% normales.  
c = sym2poly(c);  
  
% Raíces de la derivada.  
R = roots(c);  
% Solo importa la parte real (singularidades).  
R = real(R);  
  
% "Desnormaliza" las raices para eliminar aquellas que queden fuera  
% del rango de la trayectoria analizada.  
RDN = (R*sigma) + mu;  
  
% Establece límites para eliminar raíces.  
indr1 = RDN>=min(datos_Entrada(:,1));
```

```
indr2 = RDN<=max(datos_Entrada(:,1));

% Solo quedarán raíces dentro de los límites.
R = R(indr1 & indr2);

% Substituye raíces en la segunda derivada.
s = subs(sd,R);

% Obtiene los máximos y mínimos en X normalizados;
min1 = R(s>0); %Mínimos en X normalizados.
max1 = R(s<0); %Máximos en X normalizados.

% substituye máximos y mínimos en X en el polinomio para obtener sus
% Y's correspondientes.
min2 = subs(Ec_Interpolada,min1);
max2 = subs(Ec_Interpolada,max1);

% "Desnormaliza" para obtener coordenadas en X.
min1 = (min1*sigma) + mu;
max1 = (max1*sigma) + mu;

% Une máximos y mínimos en [X Y].
minT = [min1 min2];
maxT = [max1 max2];

% Convierte de números simbólicos a numeros "normales".
minT = sym2poly(minT);
maxT = sym2poly(maxT);
% Grafica mínimos en rojo y máximos en azul.
if ~isempty(minT)
    plot(minT(:,1),minT(:,2),'or');
end
if ~isempty(maxT)
    plot(maxT(:,1),maxT(:,2),'ob');
end
end
```

Herramientas para determinación de regularidad de movimiento articulado mediante el seguimiento en una secuencia de imágenes.

B.10. Función de obtención de puntos de inflexión

```
function [inflexiones] =  
pInflexion_Al(datos_Entrada,Ec_Interpolada,mu,sigma)  
  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
%                               Descripción  
% Esta función calcula los puntos de inflexión de una función matemática  
% dados los coeficientes de la función.  
%  
% Entradas:  
%  
% * datos_Entrada: Datos de entrada, donde la primera columna corresponde  
%   a las coordenadas de la trayectoria en X y la segunda columna a las  
%   coordenadas en Y.  
% * Ec_Interpolada: Vector de los coeficientes de la función creada para  
%   los datos de entrada.  
% * mu: media de los datos de entrada.  
% * sigma: desviación estándar de los datos de entrada.  
%  
% Salidas:  
%  
% Inflexiones: Matriz de coordenadas de puntos de inflexión.  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
  
%Obtención de primera y segunda derivada  
sd = diff(Ec_Interpolada,2);  
  
%Obtiene coeficientes de la primera derivada, se debe poner como [c ~]  
%para que obtenga los coeficientes en el orden que debe ser.  
[c,~] = coeffs(sd);  
  
%Convierte los coeficientes de variables simbólicas a variables  
%normales.  
c = sym2poly(c);  
  
%Raíces de la derivada.  
R = roots(c);  
  
%Solo importa la parte real (singularidades).  
R = real(R);  
  
%"Desnormaliza" las raíces para eliminar aquellas que queden fuera  
%del rango de la trayectoria analizada.  
RDN = (R*sigma) + mu;  
  
%Establece límites para eliminar raíces.  
indr1 = RDN>=min(datos_Entrada(:,1));
```

```
indr2 = RDN<=max(datos_Entrada(:,1));

%Solo quedarán raíces dentro de los límites.
R_enrango = R(indr1 & indr2);

%Substituye raíces en la ecuación original para obtener puntos de
%inflexion en(X,Y).
infX = (R_enrango*sigma) + mu;
infY = subs(Ec_Interpolada,R_enrango);
infY = sym2poly(infY);
inflexiones = [infX, infY];

%Grafica puntos de inflexión en cuadros rojos de tamaño 3.
plot(infX,infY,'sr','MarkerSize',3);
end
```

Herramientas para determinación de regularidad de movimiento articulado mediante el seguimiento en una secuencia de imágenes.

B.11. Función de obtención de velocidades

```
function [vT,vmT] = velocidad_A1(datos_Entrada,video,dist_x_pixel)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                               Descripción:
% Esta función obtiene la velocidad promedio de la trayectoria de un
% objeto en X, Y y en la dirección de la trayectoria, también obtiene una
% matriz cuyos vectores son las velocidades en X, Y y en la dirección de
% la trayectoria para cada punto.
%
% Entradas:
%
% * datos_Entrada: Datos de entrada correspondientes a las coordenadas de
%   los puntos de la trayectoria de la imagen y las posiciones en el
%   tiempo.
% * dist_x_pixel: Distancia entre pixeles en medidas reales (mm, cm,
%   etc.)
% * varargin: Se puede insertar el tiempo entre imágenes o se puede
%   calcular teniendo el número de datos y la duración del video.
%
% Salidas:
%
% * vT: Velocidades de la trayectoria punto a punto.
% * vmT: medias de las velocidades en X, Y y dirección de la trayectoria
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

X = datos_Entrada(:,1);
Y = datos_Entrada(:,2);

idx = datos_Entrada(:,end);

% Número de datos
num_Datos = length(X);

% Velocidad de cuadro del video
fr = video.FrameRate;

% Variable de tiempo por cuadro
dT = 1/fr;

% El tiempo es igual al tiempo por cuadro por el número de
% cuadros
time = idx*dT;

% Pre establecimiento de variables para mejorar velocidad de
% procesamiento.
vx = zeros(size(X));
```

```
vy = zeros(size(Y));
vdir = zeros(size(vx));
h = zeros(size(vx));

%Ciclo de cálculo de velocidades para todos los puntos de la
%trayectoria.
for i = 2:num_Datos
    difT = (time(i)-time(i-1));

    if (difT==0)
        difT = fr;
    end

    %Velocidad en X.
    vx(i) = abs((X(i) - X(i-1))/difT);

    %Velocidad en Y.
    vy(i) = abs((Y(i) - Y(i-1))/difT);

    %Velocidad en la dirección de la trayectoria.
    h(i) = sqrt((abs(X(i) - X(i-1))*dist_x_pixel).^2 +
        (abs(Y(i) - Y(i-1))*dist_x_pixel).^2);

    vdir(i) = h(i)/difT;
end

%Formación de vectores.
vT = [vx vy vdir];
vmT = mean(vT);

end
```

Herramientas para determinación de regularidad de movimiento articulado mediante el seguimiento en una secuencia de imágenes.

B.12. Función de obtención de índice de rectitud (ST)

```
function [ST] = st_A1(A)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                               Descripción:
%   Obtiene el Índice de rectitud (straightness) de una trayectoria.
%
% Entradas:
%
% * A: Puntos de la trayectoria.
%
% Salidas:
%
% * ST: Índice de rectitud.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

dE = sqrt((A(end,1)-A(1,1))^2+(A(end,2)-A(1,2))^2);
L = zeros(size(A,1),1);
for i = 2:size(A,1)
    L(i) = sqrt((abs(A(i,1)-A(i-1,1)))^2+(abs(A(i,2)-A(i-1,2)))^2);
end
ST = dE/sum(L);
end
```

B.13. Función de obtención de ángulos de dirección de objetos

```
function [vector_Theta] =
angDir_Al(video,centroides,imin,imax,sei,fondo,showim)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Descripción:
% Determina la orientación del centroide de un objeto en una secuencia
% de imágenes en los siguientes pasos:
%
% 1.- Recorta cada imagen a la zona alrededor del centroide del objeto
% detectado.
% 2.- Crea un fondo promediando todas las imágenes del video, y para
% cada imagen recortada resta el fondo de esta para dejar solamente
% el objeto de interés.
% 3.- Aplica algoritmo de ridler-Calvard para encontrar un umbral que
% binarice la imagen, dejando los pixeles del objeto de interes con
% valores de 1 y el fondo con valor de 0.
% 4.- Aplica PCA a la imagen binaria para generar un vector en la
% dirección de mayor varianza y un vector perpendicular a este, con
% estos dos vectores determina la el ángulo de dirección del objeto
% en dos dimensiones.
%
% Entradas:
%
% * video: Video a analizar.
% * Centroides: Matriz de centroides [X,Y,índices]
% * imin: Imagen desde la cual empezará el análisis.
% * imax: Imágen máxima a analizar.
% * sei: Salto o intervalo de sub muestreo entre imágenes.
% * fondo: Brinda posibilidad de establecer un fondo preguardado
% * showim: Si esta variable es diferente de cero muestra las imágenes
% con los componentes principales.
%
% Salidas:
% * vector_Theta: Vector de ángulos para cada una de las imágenes
% analizadas.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Número de imágenes del video.
N = video.NumberOfFrames;

%Validación, si no se establece fondo.
if (nargin<6)
    %Se obtiene un fondo en escala de grises del promediado de todas las
    %imagenes del video.
    fondo = subsFondo_Al(video,1,N);
    %Establece variable para no mostrar imágenes.
    showim = 0;
end
if (nargin<7)
```

Herramientas para determinación de regularidad de movimiento articulado mediante el seguimiento en una secuencia de imágenes.

```
showim = 0;
end

A = centroides;

%índices
idx = A(:,end);
%Centroides
A = A(:,1:2);
rango = 30;

% Establecimiento de variables auxiliares
cont=0;
thetaAnt = 0;
vector_Theta = zeros(1,(floor((imax-imin)/sei)));

% Ciclo de procesamiento de las imágenes
for i = imin:sei:imax

    %Busca si existe al menos un centroide en la imagen en turno
    [~,e]=find(i==idx);

    %Si existe al menos un centroide en la imagen hace todo el proceso,
    %si no, pasa a la siguiente imagen.
    if (e==1)

        %Rango a analizar alrededor del centroide.

        %Lee imagen elegida
        im = read(video,i);

        %Revisa si la imagen a analizar contiene al menos un centroide.
        cooract = A(idx==i,:);

        %Genera límites para recortar la imagen dependiendo del rango
        %establecido
        cooract(2)=480-cooract(2);
        x1 = cooract(2)-rango;
        x2 = cooract(2)+rango;
        y1 = cooract(1)-rango;
        y2 = cooract(1)+rango;

        %Validación de las distancias alrededor del objeto detectado para
        %evitar errores
        [w,h,~]=size(im);
        if(y1<1) y1=1; end
        if(y1>h) y1=h; end
        if(y2<1) y2=1; end
        if(y2>h) y2=h; end
        if(x1<1) x1=1; end
        if(x1>w) x1=w; end
        if(x2<1) x2=1; end
```

```
if(x2>w) x2=w; end

%Recorta la imagen alrededor del centroide de acuerdo al rango
%establecido
imact = im(x1:x2,y1:y2,:);

%Recorta el fondo en el mismo rango.
fondoact = fondo(x1:x2,y1:y2);

%Conversión de imagen a escala de grises y variables de tipo
%double para comparación con fondo.
imact = rgb2gray(imact);
imact = im2double(imact);
imact = medfilt2(imact);

%Implementa substracción de fondo
isf = imact - fondoact;

%Implementa técnica de Riddler-Calvard para binarizar imagen.
[Y,T] = rCalvard_Al(isf);

%Garantiza que el objeto detectado tenga valores de 1's y el
%fondo sean 0's
if (sum(sum(Y==1)))>=(sum(sum(Y==0)))
    Y = ~Y;
end

%Elimina grupos pequeños de pixeles y aplica cerradura de disco
Y=bwareaopen(Y,40);
se=strel('disk',5);
CY=imclose(Y,se);
Y=bwareaopen(CY,35); % Y1 para aplicar PCA

%Obtiene coordenadas de los pixeles del objeto y calcula su
%covarianza para determinar la dirección en la cual existe la
%mayor varianza de la imagen utilizando sus eigenvalores y
%eigenvectores (igual que en técnica de PCA).
%Esto es para conocer la posición de la gimnasta en la imagen.
[x,y]=find(Y==1);
C = [y,x];
covarianza = cov(C);
[U,S] = svd(covarianza);

%Muestra imagen en turno y grafica centroide.
if (showim~=0)
    figure(1);
    imshow(im);
end

cont = cont+1;
texto = sprintf('----- %d',cont);
set(1,'Name',texto);
hold on
plot(cooract(1),cooract(2),'*k')
```

Herramientas para determinación de regularidad de movimiento articulado mediante el seguimiento en una secuencia de imágenes.

```
%Dibuja líneas de componentes principales
drawLine(cooract,cooract-0.1 * S(1,1) * U(:,1),'-g','LineWidth', 2);
drawLine(cooract,cooract-1 * S(2,2) * U(:,2),'-g','LineWidth', 2);
drawLine(cooract,cooract+0.1 * S(1,1) * U(:,1),'-g','LineWidth', 2);
drawLine(cooract,cooract+1 * S(2,2) * U(:,2),'-g','LineWidth', 2);

%Obtiene la longitud del vector que se forma entre el centroide
%de la imagen y el primer componente principal, y utiliza este
%vector como hipotenusa para obtener el ángulo con respecto a la
%horizontal.
pf = cooract-0.1 * S(1,1) * U(:,1)';
V = pf - cooract;
V(2) = -V(2);
Vx = V(1);
Vy = V(2);

%Sentencias de decisión que toman en cuenta el ángulo de la
%imagen anterior para decidir cuál de los componentes principales
%corresponde al ángulo en turno.
if (Vx>=0 && Vy>=0)
    theta = asind(Vy/(hypot(Vx,Vy)));
    if ((thetaAnt>300) && (theta<90))
        thetaAnt = 360-thetaAnt+theta;
    end
    theta2 = theta+180;
    if (abs(theta2-thetaAnt) < abs(theta-thetaAnt))
        theta = theta2;
    end
elseif (Vx<=0 && Vy>=0)
    theta = asind(Vy/(hypot(Vx,Vy)));
    theta = 180-theta;
    theta2 = 180+theta;
    if (abs(theta2-thetaAnt) < abs(theta-thetaAnt))
        theta = theta2;
    end
elseif (Vx>=0 && Vy<=0)
    theta = asind(Vy/(hypot(Vx,Vy)));
    theta = 180 + theta;
    theta2 = 180 + theta;
    if (abs(theta2-thetaAnt) < abs(theta-thetaAnt))
        theta = theta2;
    end
end

thetaAnt = theta;

%Guarda el ángulo en turno.
vector_Theta(i) = theta;

%Si no se encontró ningún centroide pasa a la siguiente imagen;
else
    continue
end
```

```
end

%Debido a que el intervalo de imagenes "sei" podría no ser 1,
%vector_Theta puede tener valores de 0 entre cada intervalo
%ejemplo si sei=4, vector_Theta~=0 = [0 0 0 1 0 0 0 1 0 0 0 1 ...]
vector_Aux = (vector_Theta~=0);

%Vector de los ángulos de todas las imágenes analizadas
vector_Theta = vector_Theta(vector_Aux);
end
```

Herramientas para determinación de regularidad de movimiento articulado mediante el seguimiento en una secuencia de imágenes.

B.14. Función de generación de fondo promediado

```
function [fondo] = subsFondo_A1(video,L1,L2)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                               Descripción:
% Obtiene un fondo promediado en escala de grises de todas las imágenes
% de un video.
%
% Entradas:
%
% * video: video de entrada.
% * L1: Número de imagen para límite inferior de substracción de fondo.
% * L2: Número de imagen para límite superior de substracción de fondo.
%
% Salidas:
%
% * fondo: fondo promediado de todas las imagenes del video.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Características del video: Numero de cuadros y dimensiones.
L = L2-L1;
vidAlto = video.Height;
vidAncho = video.Width;
imagen_Ant = zeros([vidAlto,vidAncho]);

% Hace un preprocesamiento a cada imagen del video para después obtener
% un promedio del fondo para hacer una substracción de fondo.
for i = L1:L2
    imagen = read(video,i);
    imagen = rgb2gray(imagen);
    imagen = medfilt2(imagen);
    imagen = im2double(imagen);
    imagen = imagen_Ant + imagen;
    imagen_Ant = imagen;
end

%Fondo promediado
fondo = imagen./L;
end
```

B.15. Función de umbralado automático Ridler Calvard

```
function [Y,T] = rCalvard_Al(I)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Descripción.
% Esta función implementa el método de umbralado automático de Riddler
% Calvard, el cual selecciona un umbral T para una imagen en escala de
% grises con uno o más objetos contrastantes a partir de un algoritmo
% iterativo.
% Entradas:
% * I: imagen de la cual se obtendrá umbral T.
% Salidas:
% * Y: imagen binaria de salida.
% * T: Umbral obtenido.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

muT = mean2(I); %Se obtiene media de la imagen

[N,M]=size(I); %Tamaño de la imagen

Y=zeros(N,M); %Se crea una matriz de ceros para alocar memoria

%Ciclo para binarizar imagen con umbral muT obtenido de la media
for i=1:N;
    for j=1:M;
        if(I(i,j) > muT)
            Y(i,j)=1;
        else
            Y(i,j)=0;
        end
    end
end

ko=25; %Número de iteraciones.

%Se va actualizando el umbral T de acuerdo a la media de la imagen
%resultante en cada ciclo.
for k = 1:ko
    mu1 = mean(I(Y==1));
    mu2 = mean(I(Y==0));
    T = (mu1+mu2)/2; %Fórmula de Riddler-Calvard.
    for i=1:N;
        for j=1:M;
            if(I(i,j) > T)
                Y(i,j)=1;
            else
                Y(i,j)=0;
            end
        end
    end
end
end
```

Herramientas para determinación de regularidad de movimiento articulado mediante el seguimiento en una secuencia de imágenes.

B.16. Función de promediado de datos en coordenadas en Y

```
function [X_Uni,Y_Uni] = promY_Al(datos_Entrada)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Descripción
% Hay ocasiones en las cuales se tienen datos repetidos en X para el
% diferentes valores de Y, en estos casos es más complicado llevar a cabo
% la regresión o interpolación para generar una función que describa
% estos pares de datos. Esta Función elimina los datos repetidos de X y
% promedia los valores de Y para esas X.
%
% Entradas:
% *datos_Entrada: Datos de entrada, donde la primera columna corresponde
%                 a las coordenadas de la trayectoria en X y la segunda
%                 columna a las coordenadas en Y.
%
% Salidas:
% * X_Uni: X únicas (sin repetirse).
% * Y_Uni: Y's correspondientes a las X únicas.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

X = datos_Entrada(:,1);
Y = datos_Entrada(:,2);

%Ordena los datos de X de menor a mayor y guarda los índices en I.
[X,I] = sort(X);
%Ordena los datos de Y de acuerdo a los índices I para que queden con sus
%respectivas X.
Y = Y(I);

%Inicialización de parámetros útiles
cont= 1; % Contador para promediado de datos
Z = []; % Z contendrá los datos de Y ya promediados.
i = 2; %Variable auxiliar i para irse desplazando a lo largo de todos los
datos
% i inicial es igual a 2 porque se hace una operacion donde se necesita
i-1

while (i<=length(X)) %mientras i sea menor que el número de datos.
    aux = Y(i-1); % variable auxiliar.

    if (X(i)==X(i-1)) % Si X actual es igual a X anterior.
        while (X(i)==X(i-1)) %mientras X actual sea igual a X anterior e
i sea menor al número total de datos.
            aux = aux + Y(i); % variable aux sera igual a aux + Y actual.
            cont = cont+1; %aumenta contador puesto que ahora hay un
número más a promediar.
            i = i+1; % Aumenta i para ver si de nuevo la X siguiente será
igual a la X anterior.
        if (i>length(X)) %Si i es mayor al número de datos;
            break; %Salirse sel ciclo while y terminar proceso.
```

```
        end
    end
    else
        %Si X actual NO es igual a X anterior.
        aux = Y(i-1); %Auxiliar será simplemente el valor anterior de Y.
        cont = 1; %Establece contador a 1 puesto que no habrá promedio
    end

    VZ = aux/cont; % VZ es el promedio de la suma de los valores Y(i)
    para los cuales X(i), X(i-1),...,X(i-n) son iguales
    Z = [Z ; VZ]; %Guarda VZ en Z para ir formando el vector de salida Y
    para el cual no se repite ninguna X.
    cont = 1; %Reinicia contador para preparar cálculo del promedio
    siguiente.
    i = i+1; %Aumenta i para analizar dato siguiente (X(i+1))
    %Esta última parte evita un problema que no permitia establecer el
    %último valor de Y_Uni
    if (length(Z) < length(unique(X)) && i>length(X))
        aux = Y(end,:);
        VZ = aux/cont; % VZ es el promedio de la suma de los valores Y(i)
        para los cuales X(i), X(i-1),...,X(i-n) son iguales
        Z = [Z ; VZ];
    end
end

Y_Uni = Z; % se pasa la variable Z a Y_Uni, se llama así por Y única
X_Uni = unique(X); %Se obtienen los valores no repetidos de X

end
```

Herramientas para determinación de regularidad de movimiento articulado mediante el seguimiento en una secuencia de imágenes.

B.17. Algoritmo principal

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                               Descripción
% Este algoritmo extrae todos los parámetros de similitud de los videos
% de gimnasta y hace una comparación que obtiene un parámetro evaluación
% de similitud.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Se cargan videos a analizar
videoA = VideoReader('D:\Mis Documentos\Maestría UABC\TESIS
MAESTRIA\Saltos\Antepenultimo Salto b.MP4');
videoB = VideoReader('D:\Mis Documentos\Maestría UABC\TESIS
MAESTRIA\Saltos\Penultimo Salto b.MP4');
videoC = VideoReader('D:\Mis Documentos\Maestría UABC\TESIS
MAESTRIA\Saltos\Ultimo Salto b.MP4');

% Número de cuadros para cada video
NA = videoA.NumberOfFrames;
NB = videoB.NumberOfFrames;
NC = videoC.NumberOfFrames;

% Se cargan coordenadas de centroides antes calculadas.
A = load('APSO');
A = A.APSO;
B = load('PSO');
B = B.PSO;
C = load('USO');
C = C.USO;

% Índices de los centroides.
idxA = A(:,end);
idxB = B(:,end);
idxC = C(:,end);

% Matrices de centroides.
A = A(:,1:2);
B = B(:,1:2);
C = C(:,1:2);

% Interpolación para generación de función de comparación
[ecA, pA, ECM_propioA, muA, sigmaA] = interpolación_Al(A,19,1);
[ecB, pB, ECM_propioB, muB, sigmaB] = interpolación_Al(B,19,1);
[ecC, pC, ECM_propioC, muC, sigmaC] = interpolación_Al(C,19,1);
clc;

% Evaluación de los puntos para las funciones generadas
yevA = ev_Puntos(A,pA,muA,sigmaA,1);
yevB = ev_Puntos(B,pB,muB,sigmaB,1);
yevC = ev_Puntos(C,pC,muC,sigmaC,1);
```

```

% Obtencion de ECM's.
ECMAB = ecm_Al (B,pA,muA,sigmaA);
ECMBA = ecm_Al (A,pB,muB,sigmaB);
ECMBC = ecm_Al (C,pB,muB,sigmaB);
ECMCB = ecm_Al (B,pC,muC,sigmaC);
ECMAC = ecm_Al (A,pC,muC,sigmaC);
ECMCA = ecm_Al (C,pA,muA,sigmaA);

ECMAYB = ECMAB + ECMBA;
ECMBYC = ECMBC + ECMCB;
ECMAYC = ECMAC + ECMCA;

% Cálculo de correlaciones y covarianzas cruzadas a los puntos evaluados
[corrAB,covAB] = xcorr cov_Al (yevA,yevB);
[corrBC,covBC] = xcorr cov_Al (yevB,yevC);
[corrAC,covAC] = xcorr cov_Al (yevA,yevC);

% display('Correlaciones y Covarianzas Cruzadas:');
% display('      APS      PS      US')
corr = [max(corrAB),max(corrBC),max(corrAC)];
cov = [max(covAB),max(covBC),max(covAC)];

% Obtención de máximos, mínimos.
[minA,maxA] = pCriticos_Al (A,ecA,muA,sigmaA);
[minB,maxB] = pCriticos_Al (B,ecB,muB,sigmaB);
[minC,maxC] = pCriticos_Al (C,ecC,muC,sigmaC);

%Obtención de puntos de inflexión.
inflexionA = pInflexion_Al (A,ecA,muA,sigmaA);
inflexionB = pInflexion_Al (B,ecB,muB,sigmaB);
inflexionC = pInflexion_Al (C,ecC,muC,sigmaC);

%Alturas máximas y bajuras mínimas.
alturaA = max(maxA(:,2));
alturaB = max(maxB(:,2));
alturaC = max(maxC(:,2));

%Diferencias entre alturas y bajuras.
difaltAB = abs(alturaA-alturaB);
difaltBC = abs(alturaB-alturaC);
difaltAC = abs(alturaA-alturaC);

[EA] = find(minA(:,1)<400);
[EB] = find(minB(:,1)<400);
[EC] = find(minC(:,1)<400);
bajuraA = min(minA(EA,2));
bajuraB = min(minB(EB,2));
bajuraC = min(minC(EC,2));

difbajAB = abs(bajuraA-bajuraB);
difbajBC = abs(bajuraB-bajuraC);
difbajAC = abs(bajuraA-bajuraC);

%Comparación de velocidades.

```

Herramientas para determinación de regularidad de movimiento articulado mediante el seguimiento en una secuencia de imágenes.

```
[velsA,vmTA] = velocidad_Al ([A,idxA],videoA,1);
[velsB,vmTB] = velocidad_Al ([B,idxB],videoB,1);
[velsC,vmTC] = velocidad_Al ([C,idxC],videoC,1);

difvAB = sum(abs(vmTA-vmTB));
difvAC = sum(abs(vmTA-vmTC));

%Cálculo de Rebote.
[EA] = find(maxA(:,1)<300);
[EB] = find(maxB(:,1)<300);
[EC] = find(maxC(:,1)<300);

maxA = maxA(EA,:);
maxB = maxB(EB,:);
maxC = maxC(EC,:);

[EA] = find(maxA(:,1)>180);
[EB] = find(maxB(:,1)>180);
[EC] = find(maxC(:,1)>180);

[EAmín] = find(minA(:,1)<400);
[EBmín] = find(minB(:,1)<400);
[ECmín] = find(minC(:,1)<400);
reboteA = max(maxA(EA,2))-min(minA(EAmín,2));
reboteB = max(maxB(EB,2))-min(minB(EBmín,2));
reboteC = max(maxC(EC,2))-min(minC(ECmín,2));

difrebAB = abs(reboteA - reboteB);
difrebAC = abs(reboteA - reboteC);

% Obtención de rectitud.
STA = rectitud(A);
STB = rectitud(B);
STC = rectitud(C);

difSTAB = abs(STA-STB);
difSTAC = abs(STA-STC);

% Obtención de ángulo de dirección.
a = load('fondoAPSO.mat');
fondoA = a.fondo;
b = load('fondoPSO.mat');
fondoB = b.fondo;
c = load('fondoUSO.mat');
fondoC = c.fondo;

[angulosA] = angDir_Al (videoA,[A idxA], 320, 600, 5, fondoA, 0);
[angulosB] = angDir_Al (videoB,[B idxB], 3000, 3600, 5, fondoB, 0);
[angulosC] = angDir_Al (videoC,[C idxC], 1830, 2400, 5, fondoC, 0);

mangA = mean(angulosA);
mangB = mean(angulosB);
```

```
mangC = mean(angulosC);

difangAB = abs(mangA-mangB);
difangAC = abs(mangA-mangC);
% Vector de pesos.
%
[ECM,Correlacion,Covarianza,diferencia_de_alturas,diferencia_de_bajuras,
promedio angulos]
W = [1e-8,10,10,10,1,1,1,1,1];

% Calculo de parámetro de evaluación de similitud
regAYB = W*[ECMAYB, 1/max(corrAB), 1/max(covAB), difSTAB, difaltAB,
difbajAB, difvAB, difrebAB, difangAB]';
regAYC = W*[ECMAYC, 1/max(corrAC), 1/max(covAC), difSTAC, difaltAC,
difbajAC, difvAC, difrebAC, difangAC]';
```

Apéndice C: Tablas de datos (X,Y,t) para los videos de motor y saltos

Apéndice C

Tablas de datos (X,Y,t) para videos de saltos

C.1. Tablas de datos de videos de saltos para un sub muestreo de aproximadamente 100 imágenes

C.1.1. Tabla de datos salto A

X	Y	t	468	237	1.0417	270	200	2.2083	237	212	3.2500
845	178	0	465	244	1.0833	267	207	2.2500	236	211	3.2917
840	181	0.0417	462	248	1.1250	267	214	2.2917	237	211	3.3333
823	181	0.0833	461	261	1.1667	266	219	2.3333	237	212	3.3750
814	182	0.1250	456	271	1.2083	262	222	2.3750	238	209	3.4167
800	183	0.1667	455	279	1.2500	259	221	2.4167	237	210	3.4583
790	183	0.2083	446	291	1.2917	258	228	2.4583	236	215	3.5000
777	186	0.2500	438	300	1.3333	257	226	2.5000	236	218	3.5417
763	186	0.2917	427	306	1.3750	253	225	2.5417	235	217	3.5833
749	189	0.3333	400	335	1.4583	253	226	2.5833	235	210	3.6250
735	190	0.3750	386	327	1.5000	249	227	2.6250	235	215	3.6667
719	190	0.4167	365	323	1.5417	247	220	2.6667	234	221	3.7083
704	196	0.4583	350	312	1.5833	245	224	2.7083	235	214	3.7500
687	197	0.5000	349	299	1.6250	247	212	2.7500	236	210	3.7917
671	197	0.5417	340	287	1.6667	243	212	2.7917	237	210	3.8333
655	198	0.5833	334	272	1.7083	241	210	2.8333	236	211	3.8750
639	197	0.6250	333	256	1.7500	242	209	2.8750	233	216	3.9167
621	194	0.6667	329	250	1.7917	240	211	2.9167	235	211	3.9583
599	198	0.7083	326	230	1.8333	240	211	2.9583	236	211	4.0000
579	193	0.7500	327	226	1.8750	241	216	3.0000	236	210	4.0417
564	195	0.7917	322	223	1.9167	240	212	3.0417	234	214	4.0833
520	223	0.8750	320	210	1.9583	240	212	3.0833	235	214	4.1250
504	229	0.9167	308	204	2.0000	239	213	3.1250			
495	230	0.9583	280	186	2.0833	239	211	3.1667			
479	234	1.0000	277	185	2.1250	237	214	3.2083			

Apéndice C: Tablas de datos (X,Y,t) para los videos de motor y saltos

C.1.2. Tabla de datos salto B

X	Y	t
822	181	0
808	186	0.0498
794	183	0.0996
779	185	0.1494
762	191	0.1992
745	190	0.249
728	191	0.2988
709	194	0.3486
689	198	0.3984
669	199	0.4482
650	196	0.498
630	194	0.5478
605	194	0.5976
580	192	0.6474
559	200	0.6972
536	216	0.747
512	226	0.7968
495	230	0.8466
474	237	0.8964
460	242	0.9462
453	252	0.996
447	264	1.0458
438	279	1.0956
432	285	1.1454

427	296	1.1952
409	309	1.245
402	323	1.2948
382	335	1.3446
364	335	1.3944
347	331	1.4442
330	317	1.494
324	308	1.5438
309	286	1.6434
304	274	1.6932
299	263	1.743
290	256	1.7928
278	248	1.8426
267	235	1.8924
241	208	1.992
221	203	2.1414
214	210	2.1912
214	215	2.241
205	238	2.2908
211	219	2.2908
206	226	2.3406
202	227	2.3904
200	230	2.4402
183	209	2.6892
175	214	2.8386

171	221	2.8884
171	219	2.9382
170	226	2.988
168	222	3.0378
167	223	3.0876
164	219	3.1374
162	214	3.1872
160	228	3.2868
157	231	3.3366
153	229	3.3864
154	229	3.4362
153	231	3.486
151	225	3.5358
152	225	3.5856
151	226	3.6354
151	225	3.6852
152	225	3.735
154	224	3.7848
155	222	3.8346
155	219	3.8844
159	219	3.9342
163	220	3.984
165	234	4.0338
169	225	4.0836
171	218	4.1334

170	223	4.1832
174	221	4.233
173	225	4.2828
175	223	4.3326
177	214	4.3824
178	224	4.4322
180	221	4.482
181	224	4.5318
182	219	4.5816
180	222	4.6314
184	223	4.6812
187	221	4.731
189	216	4.7808
191	215	4.8306
191	221	4.8804
195	221	4.9302
196	224	4.98
198	225	5.0298
202	222	5.1294
205	218	5.1792
207	221	5.229

Apéndice C: Tablas de datos (X,Y,t) para los videos de motor y saltos

C.1.3. Tabla de datos salto C

X	Y	t
843	180	0
829	181	0.0583
811	180	0.1167
792	192	0.175
777	189	0.2333
757	190	0.2917
741	200	0.35
715	189	0.4083
694	195	0.4667
669	197	0.525
646	197	0.5833
620	196	0.6417
592	196	0.7
565	192	0.7583
538	213	0.8167
509	227	0.875
493	231	0.9333
473	235	0.9917
461	244	1.05
455	255	1.1083
441	280	1.225
432	293	1.2833
419	310	1.3417
402	313	1.4
382	329	1.4583
360	327	1.5167

326	299	1.6333
315	277	1.6917
308	258	1.75
306	241	1.8083
303	232	1.8667
294	222	1.925
280	210	1.9833
240	200	2.2167
233	210	2.275
229	218	2.3333
223	216	2.3917
219	220	2.45
217	220	2.5083
215	217	2.5667
211	214	2.625
209	216	2.6833
207	222	2.7417
205	218	2.8
204	219	2.8583
203	219	2.9167
203	220	2.975
202	220	3.0333
204	218	3.0917
204	218	3.15
204	219	3.2083
205	218	3.2667
206	213	3.325

208	216	3.3833
209	217	3.4417
208	220	3.5
208	218	3.5583
211	218	3.6167
209	219	3.675
212	220	3.7333
212	220	3.7917
213	220	3.85
215	217	3.9083
216	215	3.9667
214	219	4.025
211	223	4.0833
208	227	4.1417
216	220	4.2583
216	219	4.3166
214	218	4.375
215	212	4.4333
215	224	4.4916
215	214	4.55
209	224	4.6083
217	207	4.6083
217	213	4.6666
213	215	4.725
214	214	4.7833
216	214	4.8416
213	216	4.9

208	221	4.9583
212	222	5.0166
214	215	5.075
210	221	5.1333
211	223	5.1916
211	219	5.25
210	226	5.3083
211	223	5.425
212	220	5.4833
214	217	5.5416
217	218	5.6
216	221	5.6583
219	212	5.7166
224	214	5.775
225	212	5.8333
225	214	5.8916
226	216	5.95
228	220	6.0083
230	218	6.0666
233	216	6.125

C.2. Tabla de coordenadas de centroides correspondientes al target central en el mecanismo de motor de vaivén para el video de todas las lámparas encendidas

X	Y	323	169	324	169	325	172
322	169	323	169	324	170	325	172
322	170	324	169	325	170	326	173
323	169	324	169	325	171	325	173
323	169	324	170	325	171	325	174
324	169	324	170	325	172	325	174
324	169	325	170	325	173	325	174
324	170	325	171	325	173	325	175
325	169	325	172	325	173	325	175
325	170	325	172	325	174	325	174
325	171	325	173	325	175	325	173
325	172	325	173	325	175	325	173
325	172	325	174	325	175	325	172
325	172	325	174	325	174	325	172
325	173	325	174	325	173	325	171
325	174	325	175	325	173	325	171
325	174	325	174	325	172	325	171
325	175	325	174	325	172	325	170
325	175	325	174	326	172	325	170
325	175	325	173	325	171	324	169
325	174	325	172	325	170	324	169
325	173	326	172	325	169	324	169
325	173	326	172	325	170	323	169
325	173	325	171	324	169	323	169
326	172	325	170	324	169	323	169
326	172	325	170	323	169	322	169
325	171	325	170	323	169	322	170
325	171	324	169	323	169	322	171
325	170	324	169	323	169	322	171
325	169	323	169	322	169	322	171
324	169	323	169	322	171	322	170
324	169	323	169	322	171	322	169
323	169	323	169	322	171	322	169
323	169	322	170	322	171	323	169
323	169	322	171	322	170	323	169
322	169	322	171	322	170	323	169
322	169	322	171	322	170	323	169
322	169	322	171	322	170	324	169
322	170	322	171	323	169	324	169
322	171	322	170	323	170	324	170
322	171	322	170	324	169	325	171
322	170	323	169	324	170	325	171
322	170	323	170	324	170	325	172
322	169	323	169	325	170	325	172
323	169	324	169	325	171	325	173
						325	173

Apéndice C: Tablas de datos (X,Y,t) para los videos de motor y saltos

325	173
325	174
325	175
325	175
325	175
325	174
325	174
325	173
325	173
325	172
326	172
326	171
325	171
325	170
325	170
324	169
324	169
324	169
323	169
323	169
323	169
322	170
322	170
322	171
321	171
322	171
322	170
322	170
322	169
323	169
323	169
323	169
324	169

324	169
324	170
325	170
325	171
325	171
326	172
325	172
325	173
325	173
325	174
325	175
325	175
325	175
325	174
325	174
325	173
325	173
326	172
326	172
326	171
325	171
325	170
325	170
324	169
324	169
323	169
323	169
323	169
323	169
322	169
322	170
322	171
322	172

322	170
322	170
322	170
323	169
323	169
323	169
323	169
324	170
324	170
325	170
325	171
325	171
325	172
325	172
325	173
325	173
325	173
325	174
325	175
325	175
325	175
325	174
325	174
325	173
325	173
325	172
326	172
326	171
325	170
325	170
325	170
324	169
324	169

324	169
323	169
323	169
323	169
322	170
322	170
322	171
322	171
322	170
322	170
322	169
322	169
323	169
323	169
323	169
324	170
325	170
325	170
325	171
325	172
325	171
325	172
325	173
325	173
325	174
325	174
325	175
325	175
325	174
325	174
325	173