

DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE CAPTURA DE  
MOVIMIENTOS (MOCAP) DIGITAL 3D PARA ANALISIS DE SALTABILIDAD EN  
DEPORTISTAS

WILSON JAVIER LAVAO OSORIO  
JORGE LEONARDO BONILLA GÓMEZ

Trabajo de grado presentado como requisito para obtener el  
Título de Ingeniero Electrónico

DIRECTOR:  
JOSÉ DE JESÚS SALGADO PATRÓN  
INGENIERO ELECTRÓNICO

UNIVERSIDAD SURCOLOMBIANA  
FACULTAD DE INGENIERÍA  
PROGRAMA DE INGENIERÍA ELECTRÓNICA  
NEIVA - HUILA

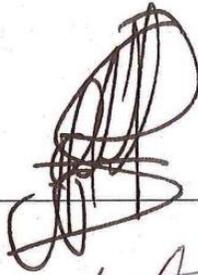
2013

DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE CAPTURA DE  
MOVIMIENTOS (MOCAP) DIGITAL 3D PARA ANALISIS DE SALTABILIDAD EN  
DEPORTISTAS

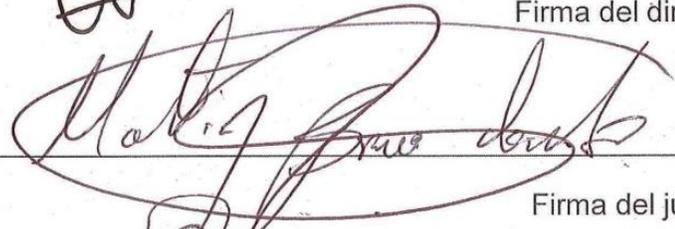
WILSON JAVIER LAVAO OSORIO  
JORGE LEONARDO BONILLA GÓMEZ

UNIVERSIDAD SURCOLOMBIANA  
FACULTAD DE INGENIERÍA  
PROGRAMA DE INGENIERÍA ELECTRÓNICA  
NEIVA  
2013

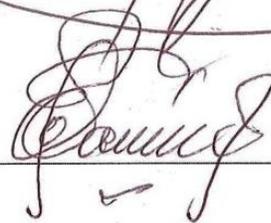
Probad, concepto de  
jurados.



Firma del director



Firma del jurado



Firma del jurado

## DEDICATORIA

En general, dedicamos este trabajo a nuestras familias. Con él, esperamos corresponder en mínima parte su abundante amor, apoyo incondicional y formación integral. En particular, a nuestros padres: Sandra y Efraín, Luz Mary y Juan Ángel por ayudarnos a convertir en mejores personas con su buen ejemplo y enseñanzas.

A mis abuelos, Leonel y Consuelo: Inspiradores de ideales.

A Valentina: La fuerza del amor.

A mis hermanos, Jhonathan, Juan Carlos, Miguel Ángel y Karina: Anhelos convertirme en motivación para unas vidas llenas de triunfo.

## **AGRADECIMIENTOS**

A Dios, por resguardar cada paso que damos, por fortalecernos en los momentos cruciales y poner en el camino a quienes nos brindaron apoyo y compañía durante nuestros años estudiantiles.

A los maestros que pacientemente sacrifican valioso tiempo y se esfuerzan grandemente en procura de un objetivo admirable: La formación profesional y humana de sus estudiantes.

Al ingeniero José de Jesús Salgado, director del proyecto, pues con su sapiencia y orientaciones logró que este barco llegase a puerto seguro.

A nuestros condiscípulos, símbolos de risas, alegrías, tristezas y esfuerzos. Tomados de las manos alcanzamos este triunfo.

A quienes presenciaron el desarrollo del presente trabajo desde sus inicios y nos ayudaron a hacerlo realidad, especialmente a los docentes Saulo Chamorro, Jaime Monje, Nelson Mendoza y Giovany Lozano, adscritos al programa de Educación Física de La Universidad Surcolombiana.

## CONTENIDO

	Pág.
INTRODUCCIÓN	20
1. CALIBRACIÓN DE LA CAMARA	22
1.1 MODELO DE LA CAMARA	23
1.1.1 Parámetros Intrínsecos	25
1.1.2 Parámetros Extrínsecos	28
1.2 PROCESO DE CALIBRACIÓN	29
1.2.1 Pasos de la calibración y el algoritmo	30
1.2.1.1 Captura	30
1.2.1.2 Detección	32
1.2.1.3 Calculo de los parámetros intrínsecos, extrínsecos y distorsión	33
1.2.1.4 Rectificación de las imágenes	34
2. CALIBRACIÓN ESTEREO	36
2.1 GEOMETRÍA EPIPOLAR	38
2.2 MATRIZ ESENCIAL	40
2.3 MATRIZ FUNDAMENTAL	41
2.4 MONTAJE IDEAL EN PARALELO DE LAS CÁMARAS	41
2.5 RECTIFICACIÓN DE LAS IMÁGENES	42
2.6 PROCESO DE LA CALIBRACIÓN ESTEREO	43
2.7. SECUENCIA DE EJECUCIÓN Y ALGORITMO	43

	Pág.
2.7.1 Obtención de la matriz T y R	44
2.7.2 Corrección de la distorsión y rectificación de las imágenes	45
2.7.3 Resultados de la calibración	46
3. SEGUIMIENTO ESTEREO	48
3.1 ESCENARIO DE CAPTURA	50
3.2 EL SALTO	51
3.3 MARCADORES	53
3.4 DESARROLLO DEL SEGUIMIENTO ESTEREO Y EL ALGORITMO	55
3.4.1 Trabajo en paralelo de las cámaras y el algoritmo	55
3.4.1.1 Rectificación de las imágenes	55
3.4.1.2 Procesamiento de los marcadores	56
3.4.1.3 Seguimiento de los puntos	58
3.4.2 Estimación de las distancias reales	59
3.4.3 Grabación del salto	61
4. VISORES 3D	62
4.1 VISOR 3D TIEMPO REAL	62
4.2 VISOR 3D TIEMPO CONTROL	64
5. RESULTADOS ESTADÍSTICOS	66
5.1 CÁLCULO DE LAS VARIABLES CINEMÁTICAS Y CINÉTICAS	66
5.1.1 Variables cinemáticas	66

	Pág.
5.1.1.1 Recorrido CG (cm)	66
5.1.1.2 Velocidad CG (m/s)	66
5.1.1.3 Aceleración CG (m/s <sup>2</sup> )	67
5.1.2 Variables cinéticas	67
5.1.2.1 Fuerza vertical (Newton)	67
5.1.2.2 Potencia mecánica (Watts)	67
5.2 ALGORITMO	67
6. HARDWARE, SOFTWARE Y MATERIALES PARA LA CAPTURA	69
6.1 HARDWARE	69
6.1.1 Cámaras web	69
6.1.2 Computador	69
6.2 SOFTWARE	70
6.3 MATERIALES	70
7. RESULTADOS	71
8. VALIDACIONES	74
9. CONCLUSIONES	78
10. TRABAJO A FUTURO	79
BIBLIOGRAFÍA	84

ANEXOS

85

## LISTA DE CUADROS

	Pág.
Cuadro 1. Salto SJ (Squat Jump).	74
Cuadro 2. Salto CMJ (Counter Movement Jump).	75

## LISTA DE GRÁFICAS

	Pág.
Gráfica 1. Recorrido	71
Gráfica 2. Velocidad	71
Gráfica 3. Aceleración	72
Gráfica 4. Fuerza	72
Gráfica 5. Potencia	73
Gráfica 6. Validación empírica de la aceleración	76
Gráfica 7. Fuerza vertical del sistema MoCap	77
Gráfica 8. Fuerza vertical plataforma Quatro Jump de Kistler	78

## LISTA DE FIGURAS

	Pág.
Figura 1. Diagrama de flujo del algoritmo de calibración	22
Figura 2. Cámara oscura	24
Figura 3. Representación real del punto principal	26
Figura 4. Distorsión radial	28
Figura 5. Damero	30
Figura 6. Variación del damero	31
Figura 7. Detección esquinas del damero	32
Figura 8. Homografía	33
Figura 9. Corrección de la distorsión radial	35
Figura 10. Diagrama de flujo del Algoritmo estéreo	36
Figura 11. Visión estéreo	38
Figura 12. Esquema de la geometría epipolar	39
Figura 13. Esquema de la geometría epipolar con matriz de rotación y vector de translación	40
Figura 14. Montaje ideal de las cámaras en paralelo	41
Figura 15. Rectificación de la imagen	42
Figura 16. Calibración estéreo	43
Figura 17. Rectificación de imágenes	47
Figura 19. Escenario de captura	49
Figura 20. Secuencia de salto Squat Jump	50

	Pág.
Figura 21. Secuencia de salto Countermovement Jump	51
Figura 22. Marcadores	52
Figura 23. Ubicación de los marcadores	52
Figura 24. Espacio de Colores	55
Figura 25. Agrupamiento de pixeles	55
Figura 26. Seguimiento Preventivo	56
Figura 27. Disparidad vs la distancia Z	58
Figura 28. Tabla para la calibración	59
Figura 29. Diagrama de flujo del seguimiento estéreo	60
Figura 30. Visor tiempo real	62
Figura 31. Diagrama de flujo del algoritmo de tiempo real	63
Figura 32. Visor 3D - Tiempo Control	64
Figura 33. Diagrama de flujo del algoritmo de tiempo control	65
Figura 34. Diagrama de flujo del algoritmo de resultados estadísticos	68
Figura 35. Cámaras web	69
Figura 36. Computador	69
Figura 37. Cámaras visión estéreo	81
Figura 38. Arquitectura básica VICON MX	82
Figura 39. Interfaz de tracker	82
Figura 40. Test de salto	85
Figura 41. Software ChronoJump	86

	Pág.
Figura 41. Software ChronoJump	86
Figura 42.Chronopic 3	87
Figura 43. Plataforma	88
Figura 44. Interfaz	89

## LISTA DE ANEXOS

	Pág.
Anexo A. Sistema ChronoJump	85
Anexo B. Guía de usuario del software	89

## GLOSARIO

**CENTRO DE GRAVEDAD:** es el punto de aplicación de la resultante de todas las fuerzas de gravedad que actúan sobre las distintas porciones materiales de un cuerpo, de tal forma que el momento respecto a cualquier punto de esta resultante aplicada en el centro de gravedad es el mismo que el producido por los pesos de todas las masas materiales que constituyen dicho cuerpo.

**DAMERO:** es un tablero de ajedrez impreso y pegado sobre una superficie plana, encargado de la calibración de las cámaras.

**FRAMES:** es un fotograma o cuadro, una imagen particular dentro de una sucesión de imágenes que componen una animación. La continua sucesión de estos fotogramas producen a la vista la sensación de movimiento, fenómeno dado por las pequeñas diferencias que hay entre cada uno de ellos.

**GEOMETRÍA EIPOLAR:** es cuando dos cámaras ven una escena 3D a partir de dos posiciones distintas, hay una serie de relaciones geométricas entre los puntos 3D y sus proyecciones sobre las imágenes en 2D que conducen a las restricciones entre los puntos de la imagen.

**GEOMETRÍA FUNDAMENTAL:** son entidades que no tienen definiciones, aunque las personas generalmente saben lo que estos conceptos significan. Son considerados conceptos apriorísticos ya que su definición sólo es posible a partir de la descripción de las características de otros elementos similares. El punto, la recta y el plano son los tres componentes fundamentales de la geometría clásica.

**MARCADOR:** son pequeñas bolas de icopor pintadas de amarillo fosforescente, ubicados lógicamente para resaltar en el traje y poder realizar el seguimiento estero.

**MATRIZ DE PROYECCIÓN:** especifica el tamaño y forma del volumen de visualización.

**MATRIZ ESENCIAL:** es una matriz 3x3 que relaciona los puntos correspondientes en imágenes estéreo.

**PIXEL:** es la menor unidad homogénea en color que forma parte de una imagen digital, ya sea esta una fotografía, un fotograma de vídeo o un gráfico.

**SALTO COUNTERMOVEMENT JUMP O CON CONTRAMOVIMIENTO:** en un movimiento descendente rápido y continuo, el individuo dobla las rodillas (fase excéntrica) hasta un ángulo de flexión de 90° (fase isométrica o acoplamiento)

manteniendo el tronco lo más próximo al eje vertical posible y desde allí genera la impulsión vertical (fase concéntrica) que lo eleva. Durante toda la fase de vuelo el atleta debe mantener sus miembros inferiores y tronco en completa extensión, hasta la recepción con la plataforma.

Es muy importante comprender que la recepción durante la caída debe ejecutarse en flexión plantar a nivel del tobillo (extensión de la articulación del tobillo) y en extensión de rodilla y cadera, para luego si generar flexión de los núcleos articulares y amortiguar el impacto generado por la masa corporal durante la caída del salto.

**SALTO SQUAT JUMP O SIN CONTRAMOVIMIENTO:** el individuo debe efectuar un salto vertical partiendo de la posición de media sentadilla (rodillas flexionadas a 90°), con el tronco erguido y con las manos dispuestas en la cintura. El individuo debe efectuar la prueba sin realizar contra movimientos hacia abajo. El salto, firme, y realizado sin la ayuda de los brazos, constituye una prueba sencilla de fácil aprendizaje y de elevada estandarización.

**SENSOR CCD - charge-coupled device:** es un circuito integrado que contiene un número determinado de condensadores enlazados o acoplados. Bajo el control de un circuito interno, cada condensador puede transferir su carga eléctrica a uno o a varios de los condensadores que estén a su lado en el circuito impreso.

## RESUMEN

El proyecto desarrolla el diseño e implementación de un sistema digital 3D de captura de movimientos, para el análisis cinemático y cinético del salto vertical en deportistas. Se fundamenta en la adquisición y tratamiento de imágenes que registran movimiento. Mediante el uso de dos cámaras web convenientemente localizadas (visión estéreo) y un puerto USB (entrada que permite a los usuarios compartir la información almacenada en diferentes dispositivos). Los autores lograron capturar ciertos marcadores o puntos de resalte jerárquicamente localizados en el cuerpo del deportista para su debida identificación. Tales puntos, comprenden un modelo biomecánico desde la cadera hasta el tobillo.

Para la captura, identificación y procesamiento de las imágenes, se desarrolló una aplicación de interfaz amigable, capaz de realizar la tarea de calibración individual de las cámaras, calibración estéreo, corrección de las imágenes y seguimiento estéreo en tiempo real, visualización 3D en tiempo real y tiempo control, análisis estadístico (análisis de las variables cinemáticas del salto). Además, una ayuda que guiará al usuario a ejecutar el software eficazmente. El programa fue implementado en lenguaje C/C++ y utiliza principalmente las librerías de visión OpenCV que son las encargadas del tratamiento de imágenes. Para terminar, se realizaron unas pruebas que comparan los resultados con otros sistemas de medición de salto vertical.

## **ABSTRACT**

This piece of research develops the design and implementation of a 3D movement capture digital system to develop cinematic and kinetic analysis of sports people vertical jump. It is based on the acquisition and treatment of images that record movement .By means of two conveniently located web cameras (stereo vision) and a USB port (entrance that allows users to share information stored on different devices), the authors were able to capture certain markers or highlighting points which were hierarchically placed in the sport person body in order to be properly identified. These points encompass a biomechanical model from hips to ankle.

In order to capture, identify and process images, it was developed a friendly interface capable to perform tasks like camera individual calibration, stereo calibration, image correction and real time stereo tracing. Such interface, also accomplishes real time and control time 3D visualization and carries out statistical analysis (analysis of jumping cinematic variables). Likewise, a set of instructions was included to help the user to take full advantage of the software. The program was implemented in C/C++ language and mainly uses OpenCV vision libraries whose main function is to treat images. Finally, some tests were conducted in order to compare the results with other vertical jump measurement systems.

## INTRODUCCIÓN

El salto se define como el movimiento que realizan los seres vivos o algunos objetos inanimados (como los robots o algunas máquinas creadas por el hombre) al impulsarse desde una superficie hacia otra a través del desplazamiento en el aire (también del agua). Para realizar tal desplazamiento, el sujeto en mención debe realizar algún tipo de fuerza la cual puede ser natural o artificialmente aplicada. Además, dependiendo de su intensidad, permitirá alcanzar distancias más o menos superiores. La acción de saltar es considerada como una de las cualidades básicas en el deportista, pues determina ciertos niveles de potencia, rapidez, coordinación, fuerza y velocidad, cuando se evalúa funcionalmente su estado físico, su proyección y su retroalimentación.

La captura de movimientos, también conocida por su equivalente inglés Motion Capture (MoCap), es el proceso de disección para el análisis y estudio de cualquier movimiento, generalmente humano. La disección consiste en la reproducción estática de diferentes instantes del movimiento, representando los pasos que se llevan a cabo.

Actualmente, se puede encontrar en el mercado local una amplia gama de sistemas de captura de movimiento para la recolección de datos y el posterior análisis de saltabilidad. Algunos de estos sistemas son eficaces y exactos, aunque limitados, pues solamente muestran dos dimensiones (2D), lo cual impide observar y analizar el movimiento del deportista desde todos los ángulos.

Los sistemas de captura de movimiento (MoCap) se consideran muy exactos teniendo en cuenta la cantidad de sensores ópticos (cámaras) que se le pueden añadir al escenario donde se capturan los movimientos. Conviene decir que, mientras más preciso sea el marcador, más exacta será la captura de movimientos del deportista lo cual mejorará la posición tridimensional.

Ahora bien, el término no se reduce a una tecnología para generar identidades realistas en películas de animación o videojuegos, pues, sus posibilidades abarcan un espectro más amplio. En este proyecto se abordará la saltabilidad de los deportistas para identificar problemas de locomoción en el momento del salto y mejorar el rendimiento.

Los autores consideran que este proyecto constituye un aporte interesante para mejorar el rendimiento de muchos deportistas en la región surcolombiana, pues, el

bajo costo económico posibilita su adquisición a laboratorios de biomédica, ligas deportivas, profesores deportivos o a personas que buscan entretenimiento. El software del proyecto está formado por varios subprogramas de los cuales cada uno cumple con una tarea específica ligada a la parte principal del software. Por tal motivo la forma de desarrollo del libro y su contexto, se realizó de la siguiente manera:

El Capítulo 1 describe la calibración individual de las cámaras, el modelo representativo de las mismas, los parámetros involucrados en el proceso de calibración, la rectificación de la imagen y por último el algoritmo con su debido diagrama de flujo que ejecuta esa parte del programa de calibración individual.

En el Capítulo 2 se desarrolla otro tipo de calibración necesaria para ajustar las cámaras entre sí, llamado calibración estéreo. Esta última comprende la visión estéreo, la geometría epipolar, el montaje frontal en paralelo, la rectificación de las imágenes y la explicación del algoritmo de calibración estéreo.

En el Capítulo 3 se detalla el seguimiento estéreo o parte central del software, al realizar la captura del movimiento del salto, procesar y hallar las distancias o posiciones de los marcadores en tiempo real para luego efectuar la grabación de los datos y el proceso que se realiza para la ejecución del algoritmo de seguimiento.

El Capítulo 4 describe los modos de visualización del modelo biomecánico de la extremidad inferior (pierna derecha) y también explica parte del algoritmo usado en la visualización.

En el Capítulo 5 se muestran las ecuaciones implementadas para obtener los valores de las variables cinemáticas y cinéticas, para luego ilustrarlas gráficamente.

En el Capítulo 6 se describe el hardware, software y materiales usados para la ejecución del proyecto.

El Capítulo 7 muestra gráficamente los resultados obtenidos de las variables cinemáticas y cinéticas del salto.

En el Capítulo 8 se realizaron las comparaciones de los resultados obtenidos con respecto a otros sistemas comerciales, en busca de la validación del sistema.

En los Capítulos 9 y 10 se muestran las conclusiones del proyecto y algunas pautas a seguir para un trabajo futuro.

# 1. CALIBRACIÓN DE LA CÁMARA

En sistemas de tratamiento de imágenes el equipo fundamental es la cámara. Por tal razón, se detallará su funcionamiento y los parámetros involucrados en ella. El proceso de calibración tiene como objetivo, conseguir una serie de parámetros que definen la forma de interactuar de la cámara con su alrededor, capturando y corrigiendo alteraciones debido al modelo de construcción e imperfecciones de fábrica.

La Figura 1, muestra el algoritmo de calibración de la cámara la cual es la primera parte de este proyecto de grado.

Figura 1. Diagrama de flujo del algoritmo de calibración

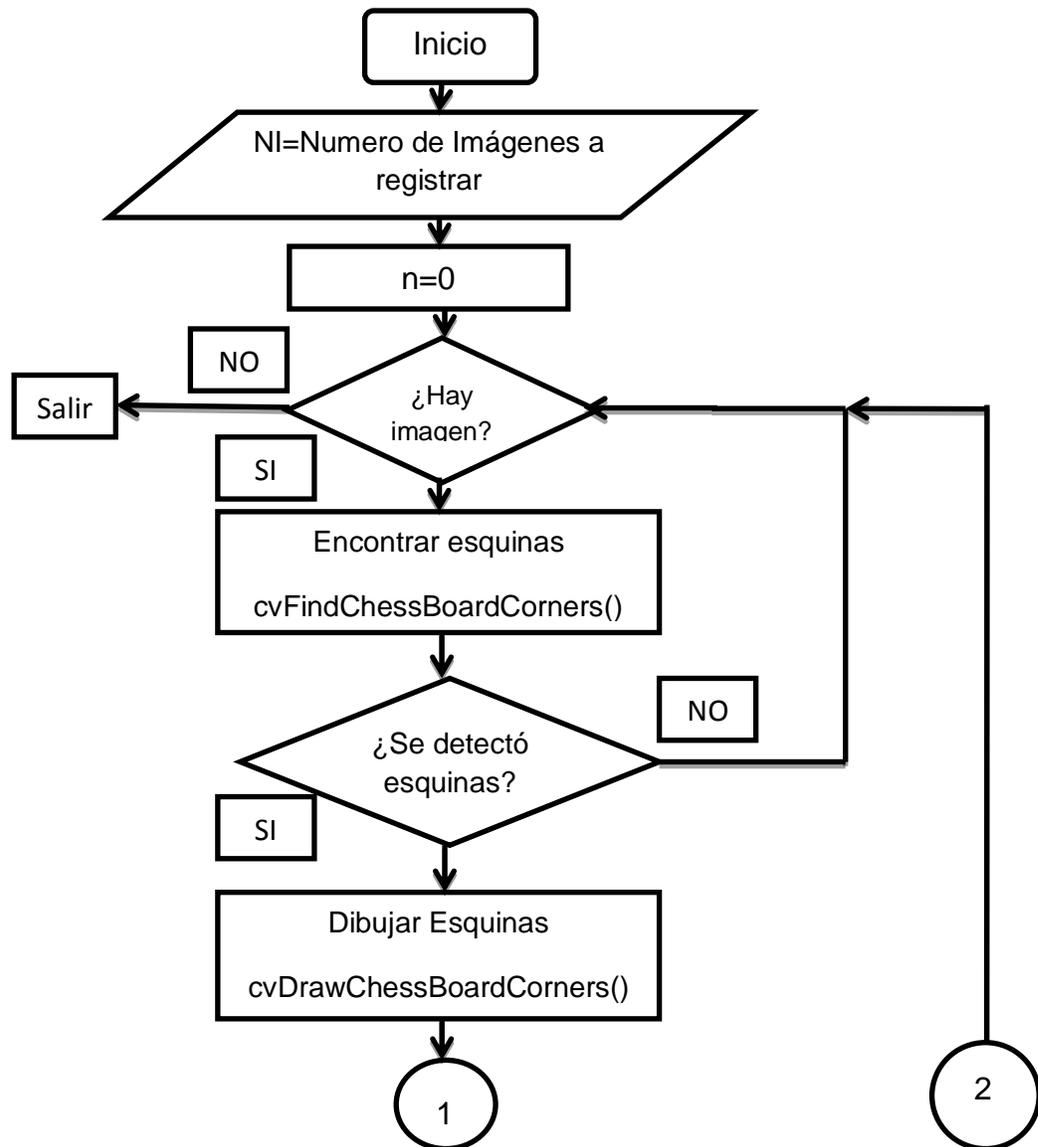
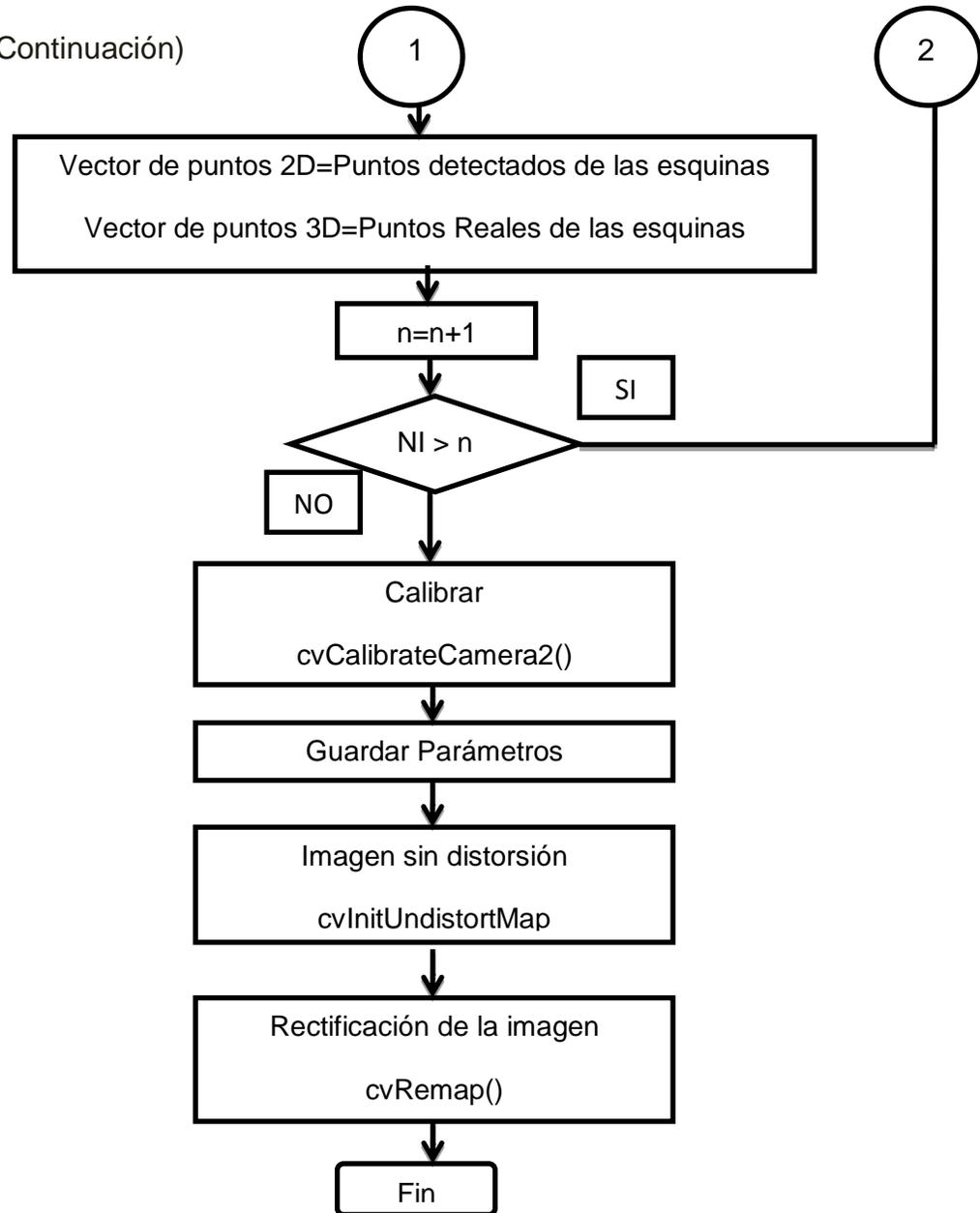


Figura 1. (Continuación)



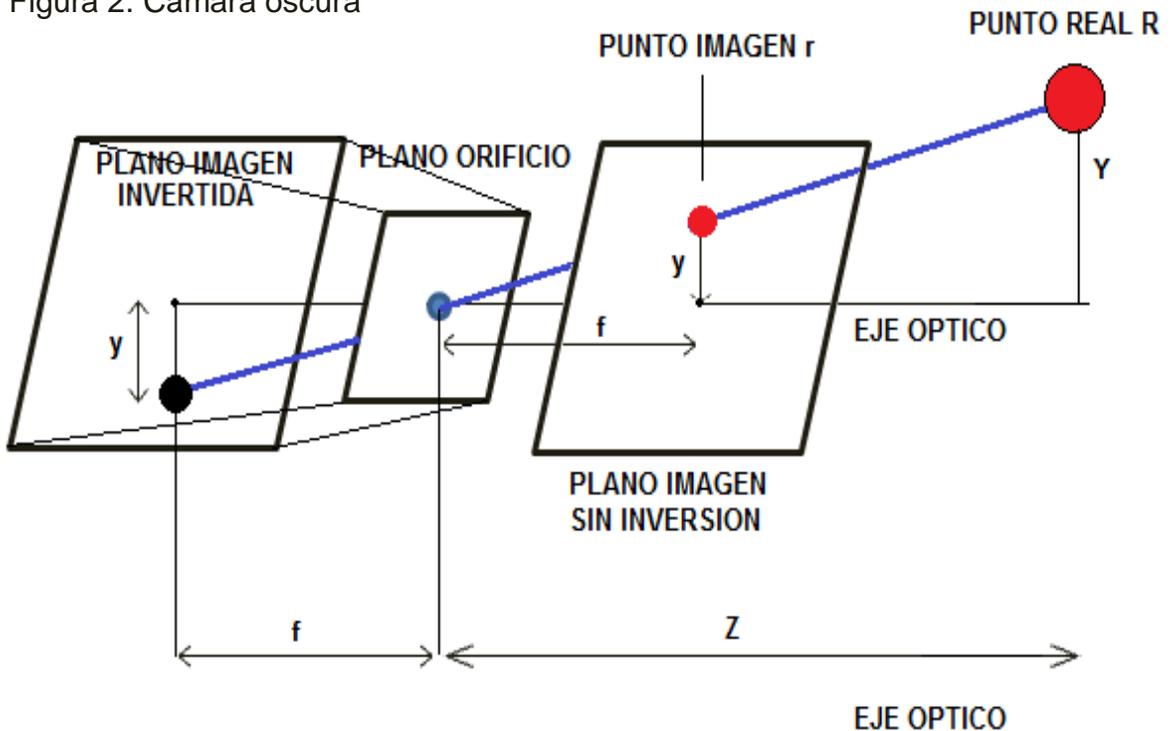
### 1.1 MODELO DE LA CÁMARA<sup>1</sup>

Uno de los modelos que explica la geometría fundamental de como un objeto capturado del mundo real se proyecta sobre un plano (por ejemplo un sensor CCD), para luego convertirse en una imagen 2D es el modelo de *pin hole*, siendo éste bastante sencillo de usar por ser un modelo lineal y basarse en el concepto de cámara oscura.

<sup>1</sup>Gary Bradski and Adrian Kaehler. Computer vision with the OpenCVlibrary. O'Reilly. (2008)

En la Figura 2 se observa como el modelo *pin hole* o cámara oscura deja incidir un único rayo de luz del objeto capturado, el cual está situado a una distancia  $Z$  desde el orificio de entrada de la luz y luego lo proyecta en el sensor CCD o plano de la imagen que se encuentra a una distancia  $f$  (distancia focal) a la abertura u orificio y éste a su vez crea una imagen 2D invertida.

Figura 2. Cámara oscura



Donde,  $y$  representa la altura del objeto en el plano de la imagen con respecto al eje óptico,  $f$  representa la distancia focal,  $Y$  representa la altura del objeto en el mundo real con respecto al eje óptico y  $Z$  representa la distancia desde el orificio al objeto real.

Ahora, para representar de forma geométrica su funcionamiento, se sitúa el plano de la imagen (imagen invertida) de una forma virtual a una distancia  $f$  pero sobre el eje  $Z$ , esto da como resultado una imagen 2D sin inversión alguna.

Al trabajar con el punto imagen  $r$  proyectada en el plano de la imagen sin inversión que es la imagen final dada por la cámara y el punto real  $R$ , con un punto central como lo es el centro óptico o punto del orificio, se encuentra una relación de triángulos, lo que facilita la obtención de los parámetros que relacionan el mundo 3D con la imágenes 2D.

En la Figura 2 se ve una correspondencia entre puntos (punto de 3D a 2D), se debe a un sistema de coordenadas homogéneo, que nos permite pasar fácilmente coordenadas con un cierto número de dimensiones a otro, sin embargo debe mantenerse la regla que la coordenada a convertir debe ser mayor en una dimensión adicional.

**R** tiene coordenadas (**X**, **Y**, **Z**) y **r(x, y)**, como el sistema de un plano imagen involucra dos dimensiones que son (**x**, **y**). Entonces, así como se implementó para la coordenada **z**, se debe trabajar para la coordenada **x** por ser coordenadas análogas trabajan de la misma manera, divergen en que la distancia focal son diferentes tanto para **y** como **x**. Aplicando esto queda trabajar con **fx** y **fy** (esta diferencia de las distancias focales es producto de la forma de las celdas receptoras de luz que conforman el sensor, porque estas no son cuadradas, si no rectangulares).

Entonces, para **y**

$$\frac{y}{fy} = \frac{Y}{Z} \quad \text{Ecuación 1}$$

Ahora para **x**

$$\frac{x}{fx} = \frac{X}{Z} \quad \text{Ecuación 2}$$

Despejando la Ecuación 1 con respecto a **y** queda:

$$y = fy \frac{Y}{Z} \quad \text{Ecuación 3}$$

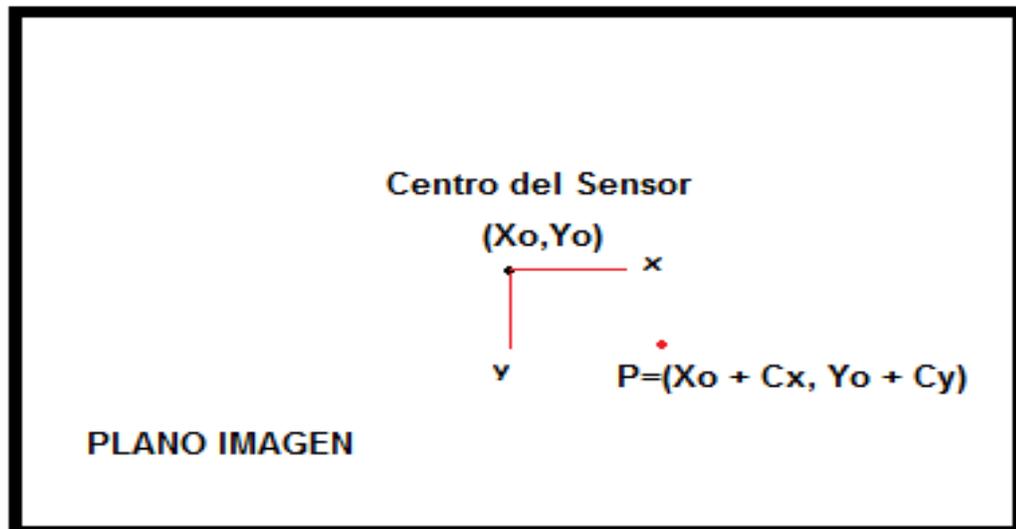
Despejando la Ecuación 2 con respecto a **x** queda:

$$x = fx \frac{X}{Z} \quad \text{Ecuación 4}$$

En la Ecuación 3 y 4 se demostró como a partir de las coordenadas reales (**X**, **Y**, **Z**) se logra adquirir de forma aproximada las coordenadas de la imagen (**x**, **y**).

**1.1.1. Parámetros Intrínsecos<sup>2</sup>.** Para alcanzar una mayor exactitud en la conversión de un punto 3D a 2D, es necesario hallar una matriz de proyección que incluya unos parámetros llamados intrínsecos obtenidos a partir de la cámara. Entre ellos se encuentra la distancia focal  $f$  ( $f_x$  y  $f_y$ ), y el centro óptico ( $C_x$ ,  $C_y$ ) que es el desplazamiento que tiene el centro de coordenadas del plano imagen respecto del punto principal. Este será nulo si el eje óptico coincide con el centro del sensor de la cámara. Sin embargo esto solo ocurre en situaciones ideales. En circunstancias reales se anexa un lente al modelo de la cámara. Se ve como el centro óptico o eje óptico se encuentra desviado del centro de la imagen generada, está posicionado en el punto donde los rayos de luz no sufren desviación alguna por la forma del lente. Ver Figura 3.

Figura 3. Representación real del punto principal



Fuente: Sistema de posicionamiento de objetos mediante visión estéreo embarcable en vehículos inteligentes

Ahora las ecuaciones 3 y 4 se transforman en,

$$y = f_y \frac{Y}{Z} + C_y \quad \text{Ecuación 5}$$

$$x = f_x \frac{X}{Z} + C_x \quad \text{Ecuación 6}$$

<sup>2</sup>Gary Bradski and Adrian Kaehler. Computer vision with the OpenCV library. O'Reilly. (2008)

Para trabajar de forma matricial se crea una tercera variable para la imagen, entonces.

$$W = Z \quad \text{Ecuación 7}$$

La **W** representa una transformación de escala para los puntos de la imagen. Como resultado se obtendrá una matriz de parámetros intrínsecos **M**, también llamada matriz de proyección de un punto 3D a la proyección de imagen 2D.

$$r = MR \quad \text{entonces, } r = \begin{bmatrix} x \\ y \\ w \end{bmatrix}, M = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad \text{y} \quad R = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad \text{Ec. 8}$$

**1.1.1.1 Distorsiones.** Alteraciones e imperfecciones en las cámaras que ocasionan datos erróneos al momento de la adquisición de las imágenes.

- La distorsión radial: la mayoría de las aberraciones producidas en las imágenes, se deben al uso de las lentes que provocan una desviación de la luz, esta aumenta en forma proporcional según la distancia al punto de incidencia con respecto al centro de la imagen, creando malformación de una manera circular, por esa razón al momento de corregirlas se emplean las siguientes ecuaciones que describen su circularidad.

$$X_c = X_s(1 + k_1 * r^2 + k_2 * r^4 + k_3 * r^6) \quad \text{Ecuación 9}$$

$$Y_c = Y_s(1 + k_1 * r^2 + k_2 * r^4 + k_3 * r^6) \quad \text{Ecuación 10}$$

**X<sub>c</sub>** = punto de la imagen en **X** corregida.

**Y<sub>c</sub>** = punto de la imagen en **Y** corregida.

**X<sub>s</sub>** = punto de la imagen en **X** con distorsión.

**Y<sub>s</sub>** = punto de la imagen en **Y** con distorsión.

En las Ecuaciones 9 y 10, **r** representa la distancia desde el centro hasta la circunferencia de aberración donde se encuentra el punto a corregir, dicho de otro modo es el radio de esa circunferencia. Las variables **k1**, **k2** y **k3** son los coeficientes que se hallarán y describirán la distorsión radial, aunque el elemento **k3** solamente es utilizado en cámaras que tiene una gran distorsión radial, como lo es el caso de cámaras *ojo de pez*, por tal motivo **k3** será igual a cero para uso de este proyecto. Ver Figura 4.

Figura 4. Distorsión radial



Fuente: <http://www.uco.es/investiga/grupos/ava/node/26>

- La distorsión tangencial: aunque ésta constituye otra causa de deformaciones de la imagen, existen algunas que aparecen con poca frecuencia. Por ejemplo, las distorsiones tangenciales ocasionadas por errores de posicionamiento del sensor fotosensible en la base de la cámara, lo cual hace que éste no quede paralelo al lente; produciendo así una ligera inclinación de la imagen generada. Conviene aclarar que en el presente trabajo no se aplicará este tipo de distorsión, por ser el resultado de la utilización de cámaras de baja calidad.

**1.1.2 Parámetros Extrínsecos<sup>3</sup>.** Son parámetros que no dependen esencialmente de la estructura interna de la cámara. Por el contrario se basa en su alrededor, es el caso de la translación y rotación, pues el objeto capturado no siempre está inmóvil, lo cual crea una perspectiva diferente de la toma de la imagen. En el modelo anterior, la cámara estaba en su posición inicial (era válido en el caso de no ser movido). Se introduce una nueva matriz la cual incluye la rotación y translación. Lo anterior se hace para modelar el sistema de forma independiente del movimiento de la cámara respecto al objeto de la cámara.

Esto es como definir la pose del objeto, gracias a que se puede obtener a partir de un mínimo de 4 puntos reales (normalmente un cuadrado con sus 4 esquinas), un ejemplo muy interesante de su uso es en la inclusión de modelos 3D con las librerías OpenGL, de esta manera se realiza la famosa realidad aumentada implementando una sola cámara. Las coordenadas manejadas por la matriz de rotación son la **R<sub>x</sub>**, **R<sub>y</sub>**, **R<sub>z</sub>** y el vector de translación son **T<sub>x</sub>**, **T<sub>y</sub>**, **T<sub>z</sub>**. Como al trabajar con plantillas 2D o planos (damero) se define a  $Z=0$ , la rotación se hace

---

<sup>3</sup>Gary Bradski and Adrian Kaehler. Computer vision with the OpenCVlibrary. O'Reilly. (2008)

nula en  $Rz$ , da como resultado una matriz de 3x2, a continuación se incluye el vector de translación para crear una matriz llamada  $W$  con una dimensión de 3x3.

Se puede hacer a  $W = [rx \ ry \ t]$ ,

Es decir que,

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = sM \begin{bmatrix} rx & ry & rz & t \end{bmatrix} \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = sM \begin{bmatrix} rx & ry & t \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \quad \text{Ecuación 11}$$

Donde  $s$  no es más que un factor de escala. Como el modelo es lineal y el sistema de coordenadas es homogéneo, del producto entre  $M$  y  $W$  resulta una matriz de homografía llamada  $H$  con una dimensión de 3x3.

Entonces,

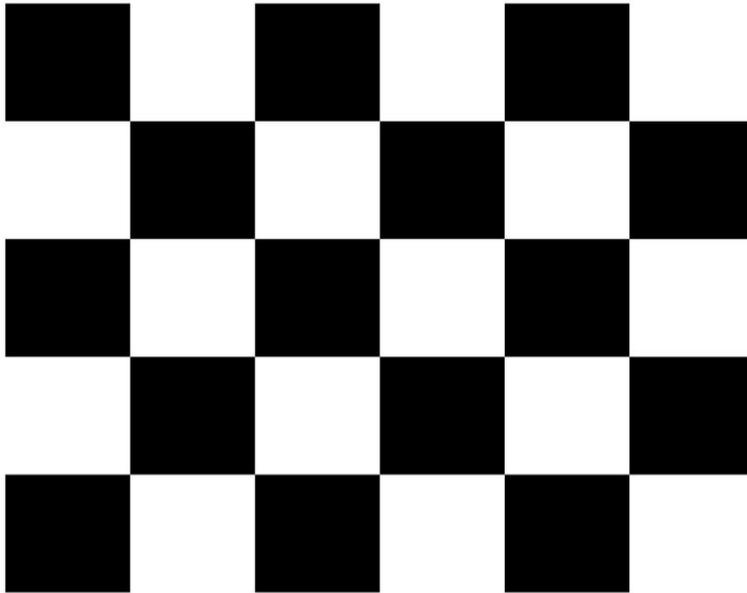
$$r' = sHR' \quad \text{Ecuación 12}$$

## 1.2 PROCESO DE CALIBRACIÓN

En el proceso de calibración, se requiere una imagen de un tablero de ajedrez el cual se imprimió y pegó sobre una superficie plana, que servirá como marcador para la obtención de los puntos reales del objeto y los proyectados en la imagen generada por la cámara. Este consta de un patrón sencillo de cuadros blancos y negros claramente diferenciados, con áreas iguales, misma longitud de ancho y de largo, capaz de dar una referencia clara y precisa de zonas o puntos a encontrar.

Para obtener una pose basado en homografía el objeto requiere un mínimo de cuatro puntos coplanares correspondientes, pero para mayor exactitud se hace uso de un gran número de puntos o esquinas contrastados como se puede visualizar en una imagen de un tablero de ajedrez y al igual que recolectar más de una imagen en la calibración, todo esto en busca de minimizar el error en los parámetros conseguidos por medio de mínimos cuadrados y un algoritmo de refinamiento iterativo. Ver Figura 5.

Figura 5. Damero



Fuente: Imágenes de la Librería OpenCV

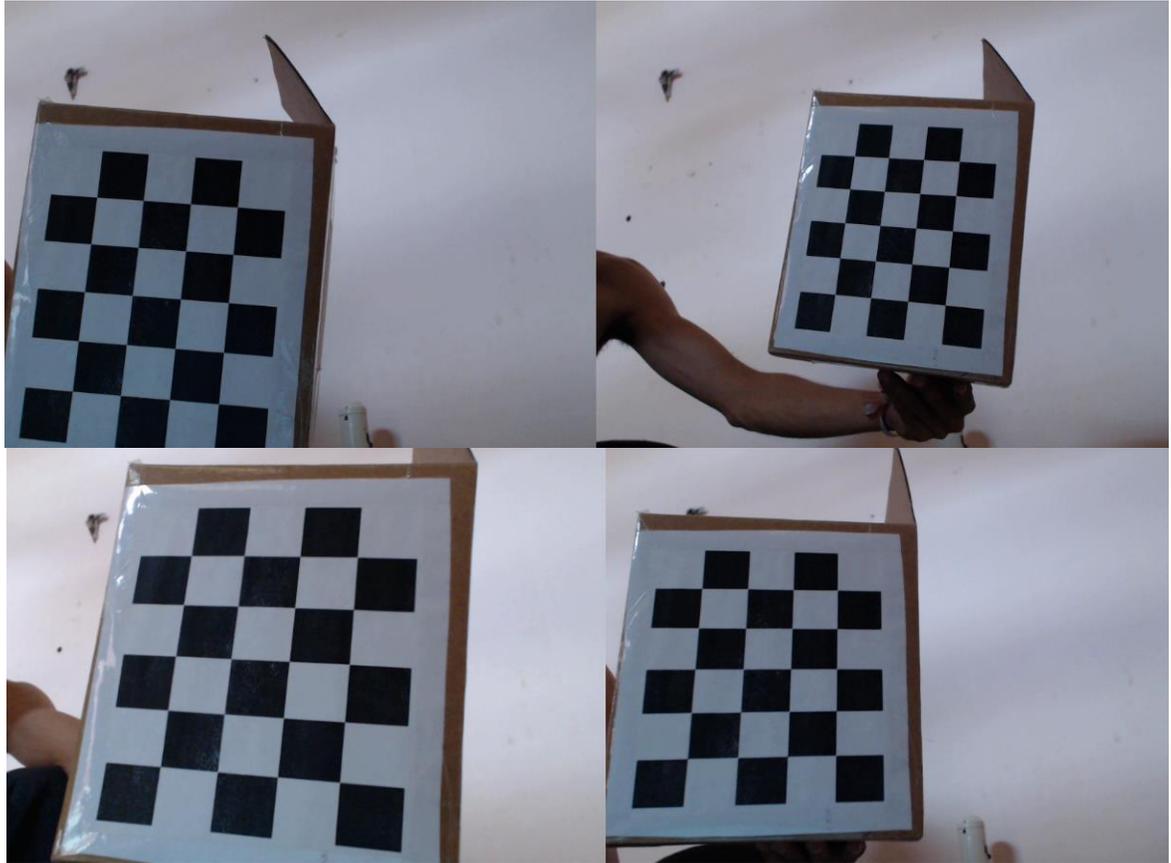
**1.2.1 Pasos de la calibración y el algoritmo.** Para calcular los parámetros intrínsecos y extrínsecos en la rectificación de las imágenes se aborda la parte más significativa o funciones importantes del código implantado en el programa de calibración.

En este proyecto se emplearán únicamente los parámetros intrínsecos como  $f$ ,  $c_x$ ,  $c_y$  y distorsión radial. Gracias a la alta calidad de las cámaras se evita la distorsión tangencial.

**1.2.1.1 Captura.** Se aplica al damero un movimiento lento, con un barrido de forma horizontal y vertical por el campo focal de la cámara para lograr un mayor alcance en aquellas zonas donde se pueden encontrar más aberraciones producto de la distorsión radial, también el ampliar el movimiento de translación y rotación crea diferentes perspectivas, logrando así una mayor precisión al momento de calcular los parámetros extrínsecos, intrínsecos y de distorsión.

En la Figura 6 se puede apreciar una serie de imágenes capturadas en la calibración, aunque no es una serie completa.

Figura 6. Variación del damero



El algoritmo usa una variable que lleva un número entero de imágenes registradas para la calibración. El proyecto cuenta con 12 imágenes para el cálculo preciso de los parámetros intrínsecos y extrínsecos. También se aplica una variable llamada **frames**, que identifica cuantas imágenes debe registrar la detección del marcador. Así habrá el intervalo suficiente para realizar un cambio de posición del damero, evitando así movimientos bruscos que causen efectos borrosos en la imagen e igualmente imprecisiones en la calibración.

Para inicializar la cámara se recurre a los siguientes comandos:

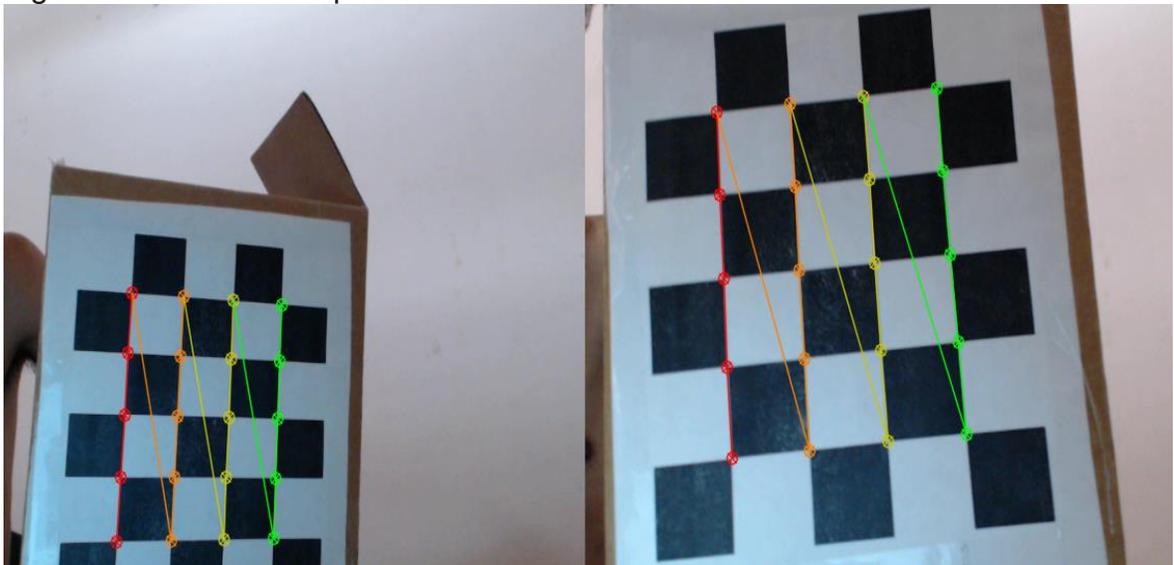
La función **cvCreateCameraCapture** (índice de la cámara) asigna e inicializa la estructura CvCapture para la lectura de un flujo de vídeo de la cámara. Actualmente, dos interfaces de cámaras se pueden utilizar en Windows: Video for Windows (VFW) y Matrox Imaging Library (MIL), y dos en Linux: V4L y FireWire (IEEE1394).

Con la función **cvSetCaptureProperty** se establece la resolución en píxeles de las imágenes, una resolución (640,480) que satisface el problema sin importar la velocidad de los cuadros adquiridos por la cámara.

Luego se utiliza el ciclo **while** para desarrollar el proceso de adquisición de imágenes, la conversión a modo escala de grises, la detección de los puntos y el dibujo de los mismos. Solo romperá el ciclo cuando cumpla con la condición de adquirir el número de imágenes necesarias registradas para la calibración.

**1.2.1.2. Detección.** Cuando se realiza el barrido del damero por todo el espacio focal de la cámara, el algoritmo busca los puntos esquinas del damero después de ser procesadas a una imagen binaria (blanco y negro), si son detectados todos los puntos se guardan en un vector 2D donde se van acumulando hasta cumplir con el número de imágenes detectadas y por cada detección se dibujan los puntos en las imágenes como se puede observar en la Figura 7.

Figura 7. Detección esquinas del damero

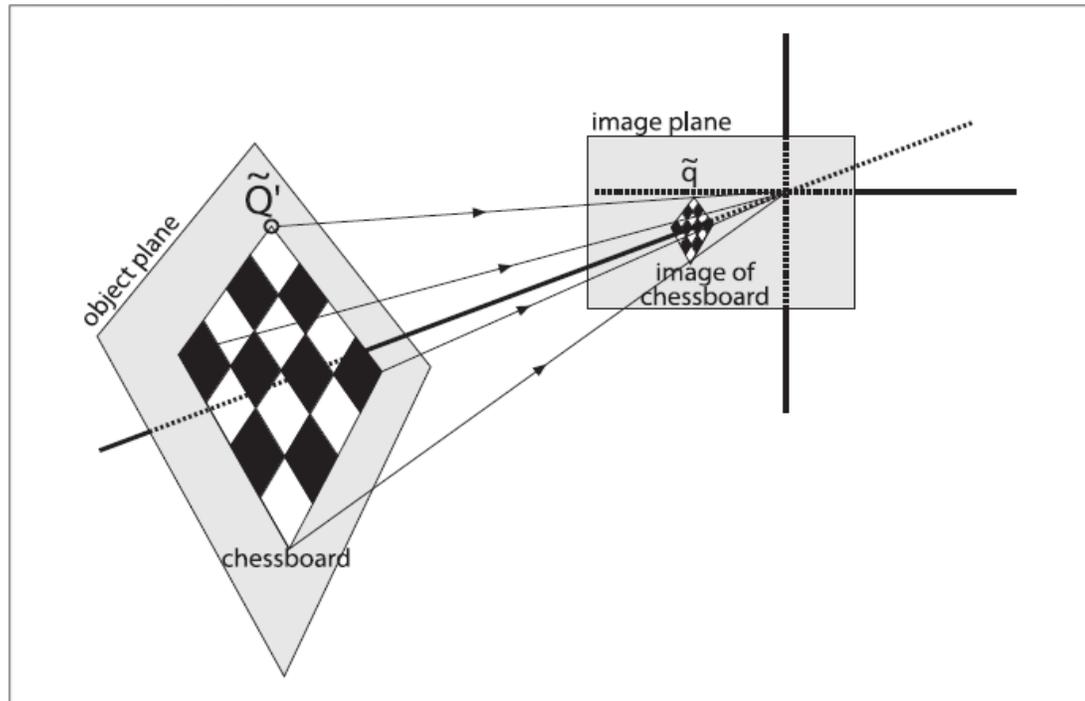


En el algoritmo se suministra el número de esquinas interiores que contiene el objeto de calibración, y se indica cuántas hay en cada lado de forma independiente. Así, la función **cvFindChessboardCorners** busca detectar en la imagen el número de esquinas indicadas y tendrá en cuenta cuál es la esquina principal o inicial. Después serán dibujadas por la función **cvDrawChessboardCorners**. Este proceso se repite según la cantidad de imágenes registradas, mientras que los puntos o posiciones se acumulan en un vector 2D el cual tiene por tamaño un número de imágenes por número de esquinas. De igual forma al vector 3D con el mismo tamaño, se le agregarán las posiciones reales de las esquinas del damero en forma sucesiva pero en relación a la esquina principal (se puede trabajar como una unidad la distancia entre esquinas y el vector **Z** que equivale a cero).

### 1.2.1.3. Cálculo de los parámetros intrínsecos, extrínsecos y distorsión.

Ahora que se tiene una relación entre los puntos de las imágenes capturadas y los puntos reales del damero, se pueden hallar los parámetros necesarios para la rectificación de las imágenes mediante la resolución de una homografía entre puntos. Lo anterior se puede apreciar en la Ecuación 12 y observar en la Figura 8.

Figura 8. Homografía



Fuente: O'Reilly Learning OpenCV, Computer Vision with the OpenCV Library

Para procesar la relación, inicialmente se necesitan los parámetros intrínsecos y extrínsecos, sin ninguna distorsión. Después de encontrar solución a estos parámetros la función realiza la operación para hallar la distorsión en forma no lineal, sino iterativa. La función encargada del proceso es **cvCalibrateCamera2** que utiliza el método de Zhang<sup>4</sup> para hallar los parámetros intrínsecos y extrínsecos, y el método de Brown<sup>5</sup> para las distorsiones.

Una de las grandes ventajas de utilizar los dos métodos anteriores en la detección de los parámetros de la cámara, es su implementación en las librerías gratuitas de OpenCv que tiene una gran eficiencia y acogida ante librerías de software comercial, también su rápida y fácil forma de implementar, la gran veracidad de los

---

<sup>4</sup>Zhengyou Zhang. A flexible new technique for camera calibration.(1998)

<sup>5</sup>Gary Bradski and Adrian Kaehler.Computer vision with the OpenCVlibrary.O'Reilly. (2008)

parámetros obtenidos y su sencillez al momento de la calibración por utilizar una plantilla bidimensional con un patrón muy sencillo que la hace muy móvil a diferencia de otros métodos que utilizan plantillas 3D.

**1.2.1.4. Rectificación de las imágenes.** Es el momento final del programa de calibración, cuando se guardan en archivos .XML los parámetros intrínsecos y de distorsión para un uso posterior evitando así la repetición del proceso, pues, una vez obtenidos los parámetros, la estructura interna de la cámara no cambia, al menos en casos especiales cuando a la cámara se le aplique un enfoque diferente de tipo hardware o software.

Para confirmar el éxito de la calibración, es necesario observar las imágenes rectificadas. Para ello se usa la función **cvInitUndistortMap** la cual ofrece dos matrices previamente creadas y que al ser utilizadas con las imágenes a rectificar (imágenes adquiridas desde la cámara) en la función **cvRemap**, muestran los resultados de la rectificación. Es importante resaltar que es el usuario por medio de la observación quien determina la calidad de la calibración, en caso de un mal resultado se debe repetir con mayor cuidado el proceso.

Procedimiento para la rectificación:

**C:** void cvRemap (CvArr const \* **src**, CvArr \* **dst**, CvArr const \* **MapX**, CvArr const \* **mapy**, int **flags** = CV\_INTER\_LINEAR + CV\_WARP\_FILL\_OUTLIERS, CvScalar **fillval** = cvScalarAll (0))

**Python:** cv. Reasignar (src, dst, MapX, mapy, banderas = CV\_INNER\_LINEAR + CV\_WARP\_FILL\_OUTLIERS, fillval = (0, 0, 0, 0)) → Ninguno

Parámetros:

- **src** - Fuente imagen.
- **dst** - imagen de destino. Tiene el mismo tamaño que map1 y del mismo tipo que el original.
- **map1** - El primer mapa de cualquiera (x, y) puntos o sólo los valores x que tienen la CV\_16SC2 tipo, CV\_32FC1 o CV\_32FC2. Ver convertMaps () para obtener más información sobre la conversión de una representación de punto flotante a punto fijo para la velocidad.
- **map2** - El segundo mapa de valores y tienen la CV\_16UC1 tipo, CV\_32FC1, o ninguno (mapa vacío si map1 es (x, y) puntos), respectivamente.
- **interpolación** - Interpolación método (ver redimensionar () ). El método INTER\_AREA no es compatible con esta función.
- **borderMode** - método de extrapolación del pixel (ver borderInterpolate () ). Cuando borderMode = BORDER\_TRANSPARENT, significa que los píxeles de la imagen de destino que corresponde a los "valores atípicos" en la imagen de origen en que no son modificados por la función.

- **borderValue** - valor utilizado en caso de una frontera constante. De forma predeterminada, es 0.

Figura 9. Corrección de la distorsión radial



Fuente: O'Reilly Learning OpenCV, Computer Vision with the OpenCV Library

## 2. CALIBRACIÓN ESTEREO

A lo largo de su evolución del ser humano ha hecho uso de una visión estéreo la cual se entiende como el proceso visual que permite una percepción de profundidad. Gracias a la separación de los ojos, a medida que la persona crece o desarrolla, aprende a interpretar la información de las imágenes adquiridas. Este hecho se debe a la eficiente y compleja forma de aprender y trabajar del cerebro humano. El presente proyecto se basa en el uso de la visión estéreo; por ello a lo largo de este capítulo se expondrá la manera como se ha emulado este fenómeno mediante un sistema de captura de imágenes. De igual manera, Los autores explicaran como se ha utilizado la información para triangular objetos. No obstante, pero para lograr la triangulación se debe recurrir a la calibración estéreo, operación que ayudará a comprender como están vinculadas o relacionadas dos cámaras entre sí.

La figura 10, muestra los pasos específicos para la calibración estéreo mediante un diagrama de flujo.

Figura 10. Diagrama de flujo del Algoritmo estéreo

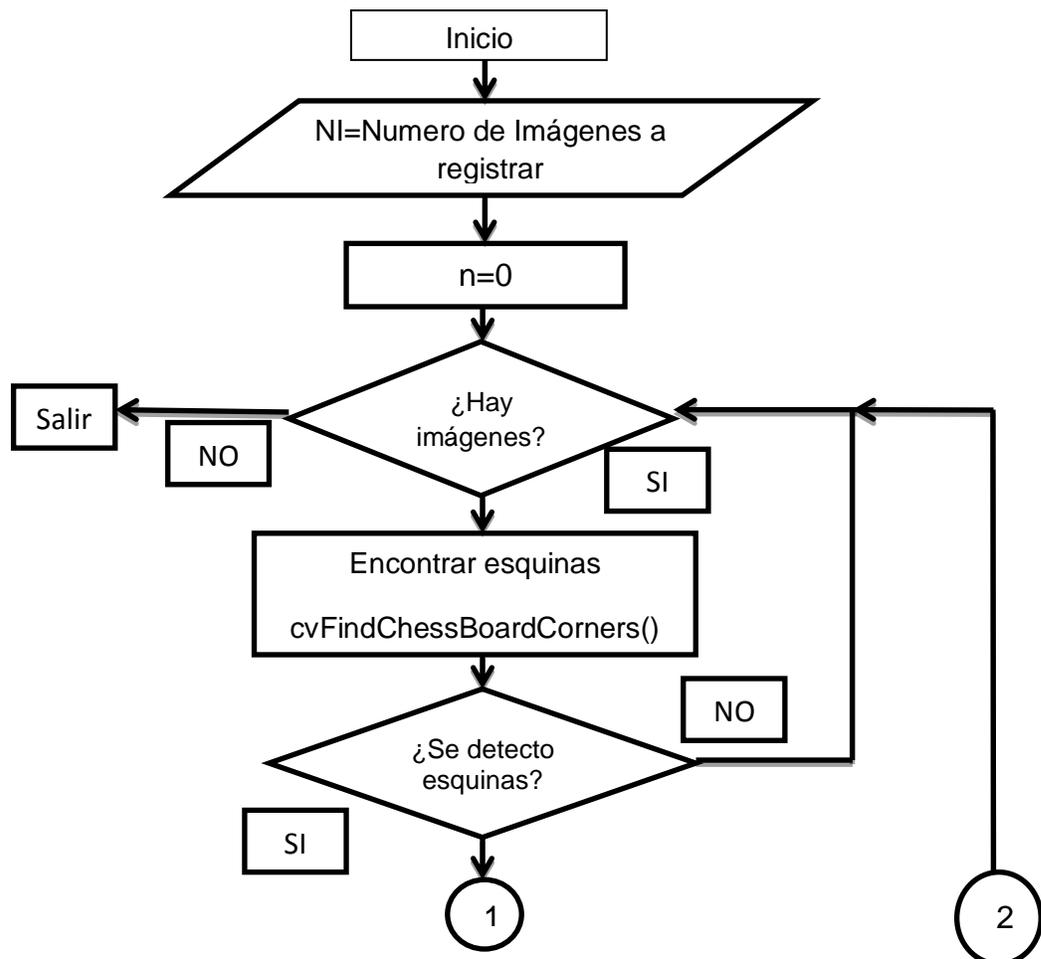
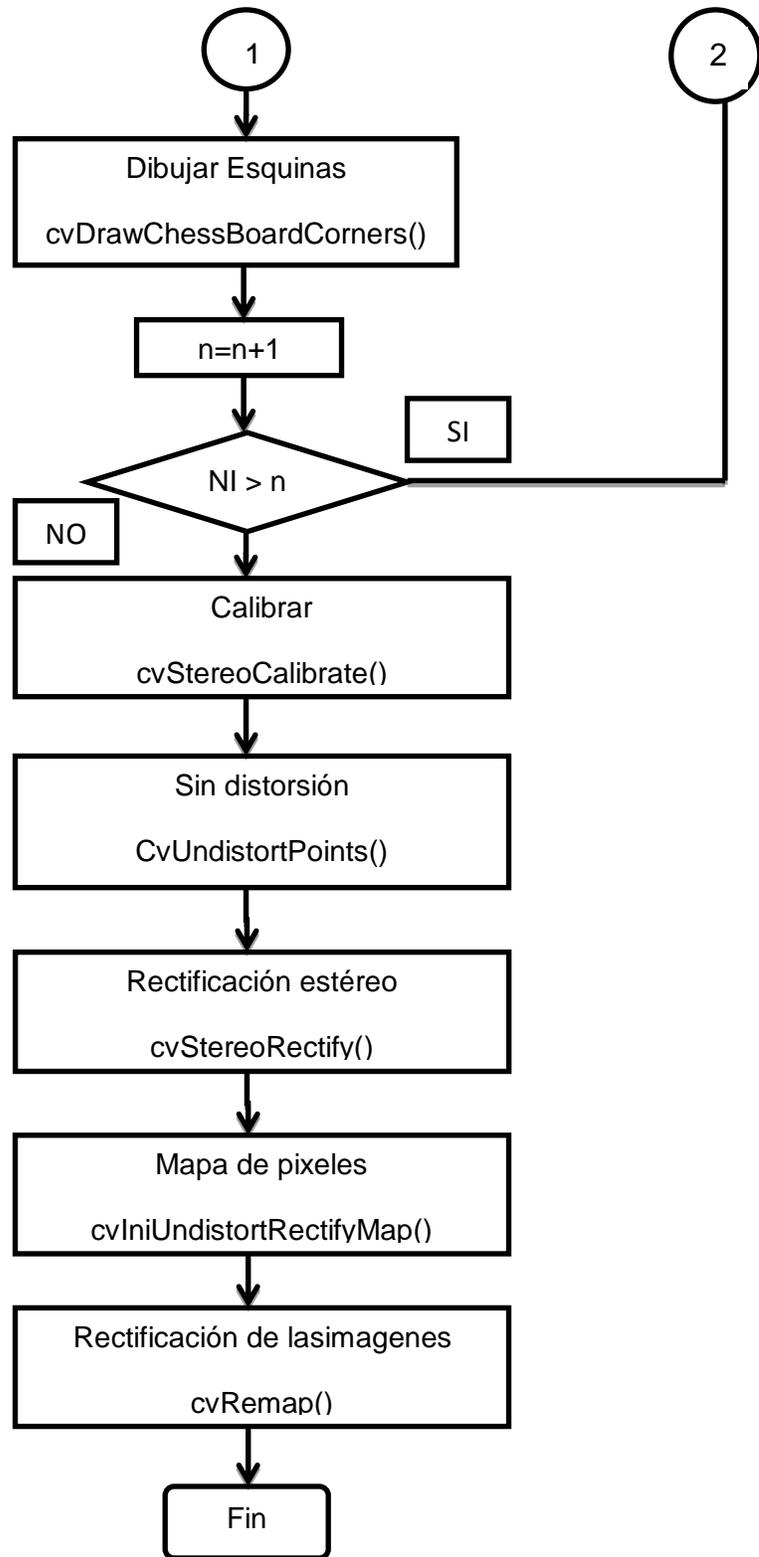


Figura 10. (Continuación)

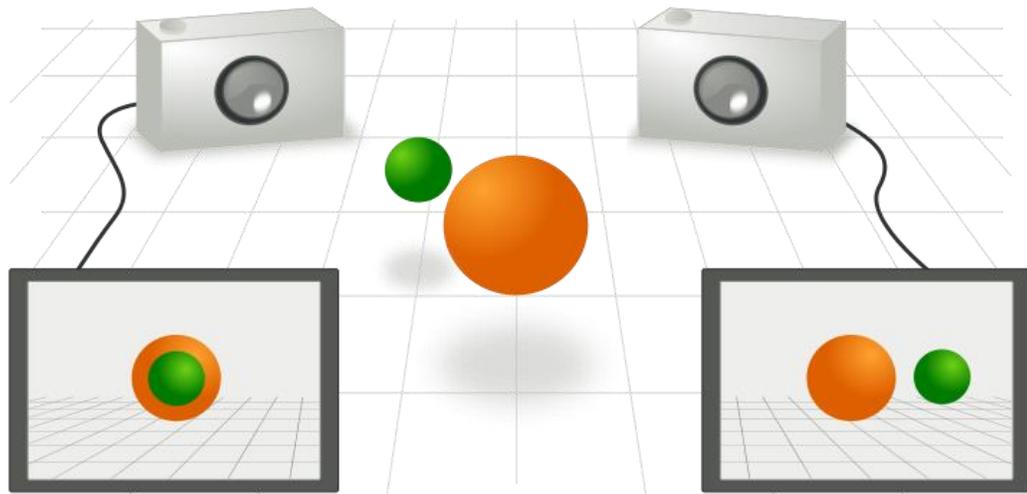


## 2.1. GEOMETRIA EPIPOLAR<sup>6</sup>

Para estimar las coordenadas 3D de un punto físico, se debe recurrir al uso de sistemas estéreos que se componen de un par de cámaras, generando sus respectivos planos de imagen. A diferencia del uso de una sola cámara, esta doble información hace posible la reconstrucción de un punto 3D. Para describir el sistema se recurre al uso de una geometría especial llamada geometría epipolar.

En la Figura 11 se puede observar como cada cámara genera una imagen diferente por estar en distintas posiciones, este desfase de posición de una con respecto a la otra, crea una relación tridimensional.

Figura 11. Visión estero



Fuente: [http://en.wikipedia.org/wiki/File:Aufnahme\\_mit\\_zwei\\_Kameras.svg](http://en.wikipedia.org/wiki/File:Aufnahme_mit_zwei_Kameras.svg)

Si se observa la Figura 12 se analiza lo siguiente:

$X$  es un punto físico del espacio, que tiene una proyección en cada una de las cámaras, tanto izquierda como derecha, si se coge una recta que una  $X$  con  $O_L$  igual con  $O_R$  que son los centro ópticos, se ve como el corte entre las rectas y el plano imagen crean un punto de proyección en cada imagen llamados  $X_L$  y  $X_R$  respectivamente, ahora se unen los dos centros ópticos mediante una línea imaginaria, se llama a  $e_L$  al punto en el que dicha recta corta el plano imagen de

---

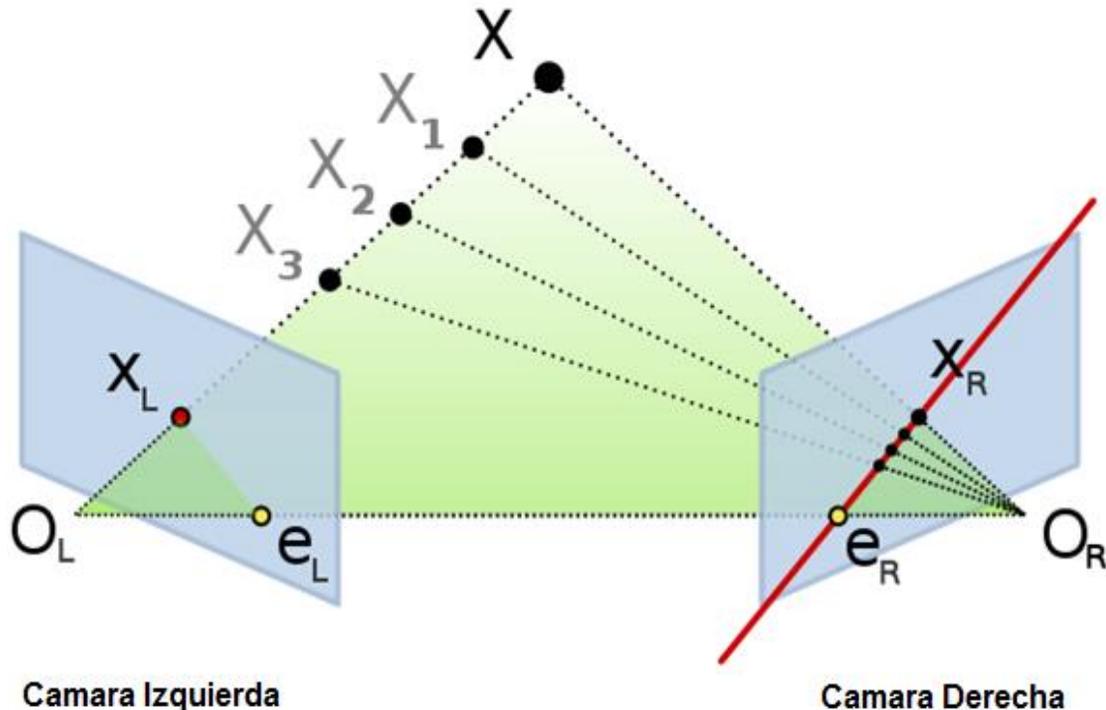
<sup>6</sup>Blanes F.; Jiménez L.M.; Puerto R.; Neco R.P.; Reinoso O." reconstrucción tridimensional de escenas con un par estereoscópico de cámaras". (2008).

la primera cámara, y  $e_R$  su contrapartida. Dicho en otras palabras, los puntos epipolares son la proyección en el plano de imagen de la otra cámara.

Luego, el plano formado por los puntos  $O_L$ ,  $O_R$  y  $X$ , se llama plano epipolar.

La rectas  $e_L X_L$  y  $e_R X_R$  son las llamadas líneas epipolares **LL** y **LR**, producto del corte entre el plano epipolar y el plano imagen de las cámaras.

Figura 12. Esquema de la geometría epipolar



Fuente: [http://en.wikipedia.org/wiki/File:Epipolar\\_geometry.svg](http://en.wikipedia.org/wiki/File:Epipolar_geometry.svg)

Se puede observar en la Figura 12, como los puntos  $X$ ,  $X_1$ ,  $X_2$  y  $X_3$  que están en la recta  $O_L X$ , tienen una misma proyección en el plano imagen izquierdo, pero en la imagen de la derecha va cambiando en forma lineal su proyección, se puede concluir que esta línea de cambio no es más que la línea epipolar **LR**, funciona de forma análoga a su contraparte. Por tanto si sabe que el punto que se busca está dentro de una recta, no hay que hacer una búsqueda en toda la imagen a la hora de buscar correspondencias entre puntos de las dos imágenes, lo cual reduce el problema de la correspondencia en una sola dimensión.

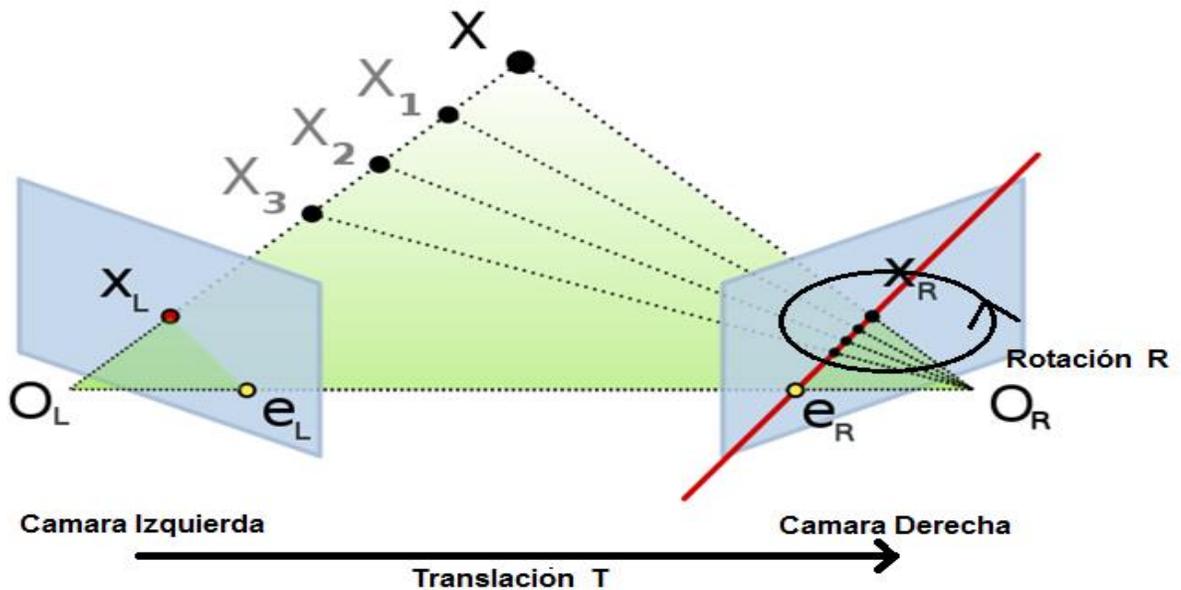
Después de explicado lo anterior, se enuncia algunas propiedades de este sistema:

- Todos los puntos 3D que pueden ser vistos por una cámara están contenidos dentro de un plano epipolar, que a su vez intercepta el plano de imagen creando una línea epipolar.
- Dado el punto de vista de un punto 3D en un plano imagen, la vista de dicho punto en el otro plano de imagen se debe corresponder estar dentro de la línea epipolar. Esto es conocido como la restricción epipolar.
- En el caso de haber más de un punto y aparecen en un orden en un plano imagen de una cámara, aparecerán en el mismo orden en el otro, al menos que se presente alguna oclusión.

## 2.2 MATRIZ ESENCIAL<sup>7</sup>

La matriz esencial tiene toda la información sobre la posición de una cámara respecto a la otra, es decir, la translación  $T$  y la rotación  $R$  relativa. Ver Figura 13. Esta matriz contiene la relación espacial o física entre las dos cámaras, no suministra información alguna sobre parámetros internos de las cámaras. Esta matriz permite tomar el punto de una imagen y proyectar una línea en la otra imagen, la cual se encontrará la proyección del otro punto.

Figura 13. Esquema de la geometría epipolar con matriz de rotación y vector de translación



Fuente: [http://commons.wikimedia.org/wiki/File:Epipolar\\_geometry.svg](http://commons.wikimedia.org/wiki/File:Epipolar_geometry.svg)

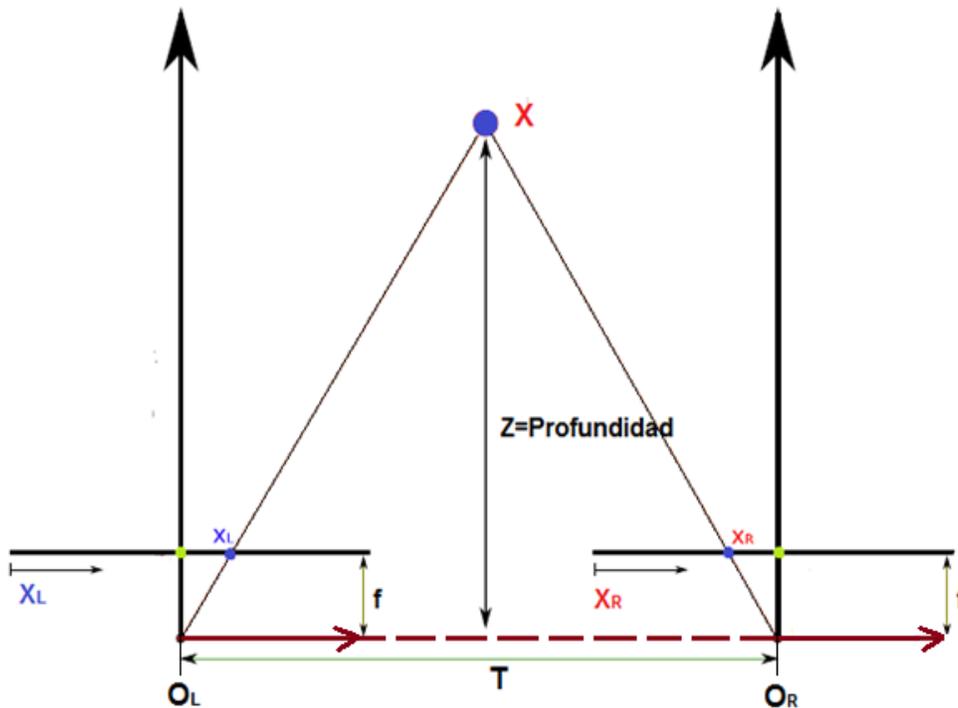
<sup>7</sup>Blanes F.; Jiménez L.M.; Puerto R.; Neco R.P.; Reinoso O." reconstrucción tridimensional de escenas con un par estereoscópico de cámaras". (2008).

### 2.3. MATRIZ FUNDAMENTAL<sup>8</sup>

A diferencia de la matriz  $E$ , la matriz  $F$  es más completa, al contar con la combinación de los parámetros externos que son la translación y rotación entre sí de las dos cámaras (matriz  $E$ ) y contar con los parámetros de proyección (parámetros intrínsecos) de cada cámara. Al igual cumple la relación para hallar un punto de una imagen con respecto a la otra con sus líneas epipolares.

### 2.4. MONTAJE IDEAL EN PARALELO DE LAS CAMARAS

Figura 14. Montaje ideal de las cámaras en paralelo



Fuente: O'Reilly Learning OpenCV, Computer Vision with the OpenCV Library

Este simple montaje idealizado, permite hallar la distancia de un punto en el espacio real de una manera más fácil, la forma del diseño comprende en colocar las dos cámaras de forma paralela, de modo que sus ejes ópticos queden completamente en paralelos con respecto a cualquier coordenada, al igual que deben quedar los planos imagen en forma coplanares y debe mantener la misma altura de sus centros ópticos. Ver Figura 14.

El esquema de la Figura 14 recrea una forma fácil de encontrar los puntos 3D por medio de la triangulación, pero no deja de ser un montaje ideal. En la vida real, emular el diseño es bastante complicado porque cualquier diferencia minúscula en

<sup>8</sup>Gary Bradski and Adrian Kaehler. Computer vision with the OpenCVlibrary. O'Reilly. (2008)

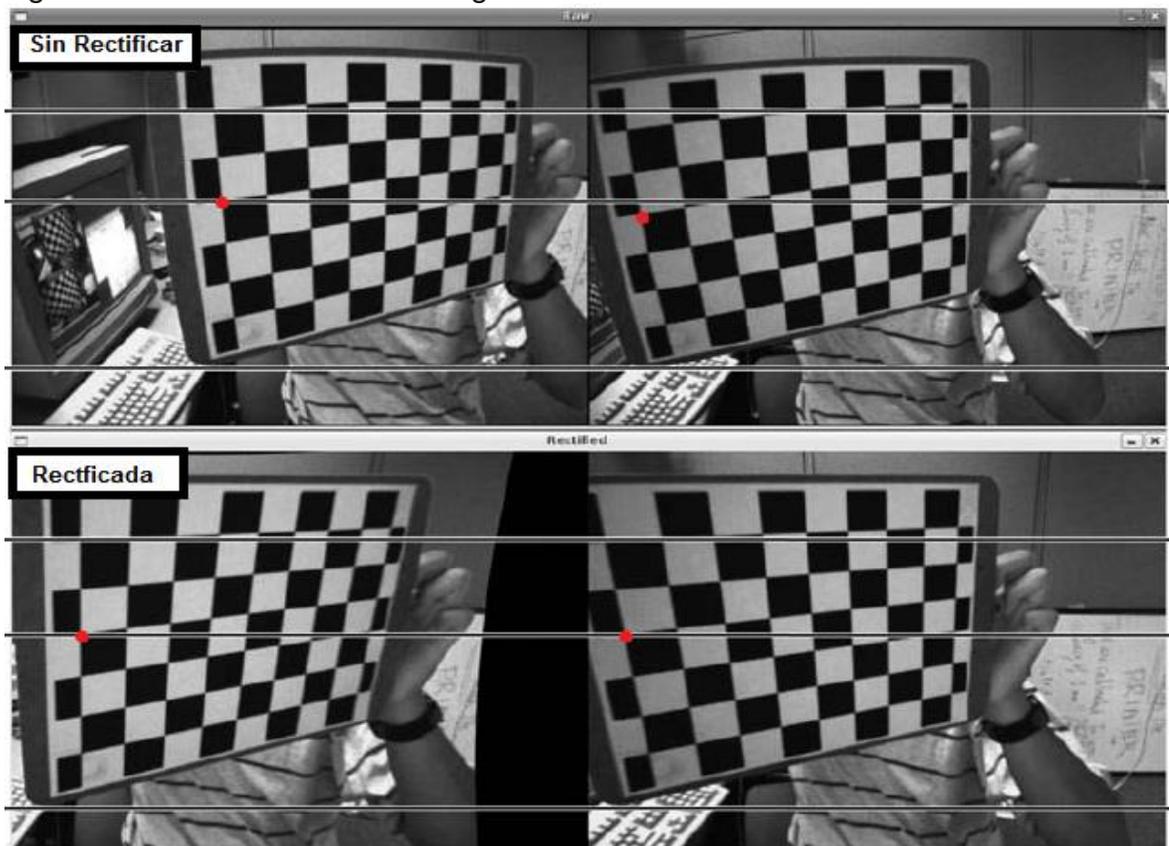
cuestión de la altura del centro óptico y desviación del eje óptico producirá un error en el cálculo de los puntos del espacio.

## 2.5. RECTIFICACIÓN DE LAS IMÁGENES

La rectificación de las imágenes es un camino para lograr ese montaje ideal explicado en el apartado anterior, esta no se apega a lo físico, sino que es causada por una buena programación que al final del registro de las imágenes se presentan como si hubiesen sido adquiridas de forma paralela ideal, pero con un poco menos de información sobre el campo visual. Ver Figura 15.

Como consecuencia se obtiene mayor eficiencia al momento de encontrar los puntos proyectados, gracias a que el problema de correspondencia ha quedado limitado a una sola dimensión, al estar siempre a una misma altura.

Figura 15. Rectificación de la imagen



Fuente: O'Reilly Learning OpenCV, Computer Vision with the OpenCV Library

## 2.6. PROCESO DE LA CALIBRACIÓN ESTEREO

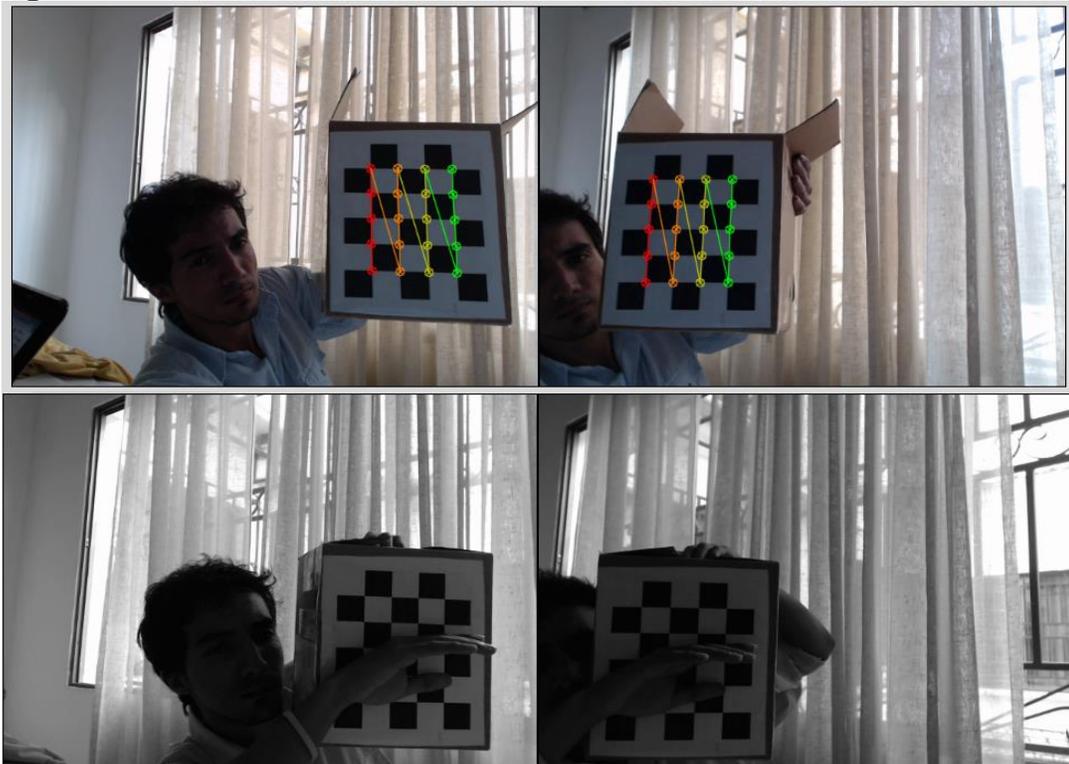
Los objetivos principales de la calibración estéreo son calcular unas matrices capaces de rectificar las imágenes para implementar el sistema en forma paralela explicado en el apartado anterior. Rectificar las distorsiones y encontrar la matriz de reproyección mutua llamada  $Q$ , permite proyectar puntos 2D de las imágenes al mundo real, consiguiendo así, establecer sin ambigüedad una correspondencia entre los puntos 3D y píxeles de las imágenes. Lo último será utilizado en un procesamiento posterior.

## 2.7. SECUENCIA DE EJECUCIÓN Y ALGORITMO

La forma de ejecutar la calibración estéreo es muy parecida a la calibración individual. Ésta última difiere de la primera en el número de cámaras usadas. Además, en el momento de la captura requiere adquirir imágenes de las dos cámaras de forma secuencial, lo cual produce un cambio en la detección de los puntos del damero. Esto implica buscar puntos en las dos imágenes y crear otro vector 2D para poder guardar los puntos identificados de la segunda imagen. Es conveniente tener en cuenta el número de imágenes a registrar.

La Figura 16 muestra la visualización de las imágenes al momento de ser detectados los puntos del damero y en caso contrario.

Figura 16. Calibración estéreo



**2.7.1 Obtención de la matriz T y R.** Se aplica análogamente a la calibración individual el mismo tipo de algoritmo para la detección de las esquinas del damero con la diferencia que son dos cámaras las usadas en la calibración estéreo.

Para obtener la matriz **T** y **R** de la calibración estéreo se utiliza la siguiente función:

La función **cvStereoCalibrate** tiene los siguientes parámetros:

- **objectPoints ( IntPtr )** La matriz conjunta de los puntos del objeto,  $3 \times N$  o  $N \times 3$ , donde  $N$  es el número total de puntos en todas las vistas.
- **imagePoints1 ( IntPtr )** La matriz conjunta de puntos de imagen correspondientes en los puntos de vista de la primera cámara,  $2 \times N$  o  $N \times 2$ , donde  $N$  es el número total de puntos en todas las vistas.
- **imagePoints2 ( IntPtr )** La matriz conjunta de puntos de imagen correspondientes en las vistas desde la segunda cámara,  $2 \times N$  o  $N \times 2$ , donde  $N$  es el número total de puntos en todas las vistas.
- **pointCounts ( IntPtr )** Vector que contiene los números de puntos en cada vista,  $1 \times M$  o  $M \times 1$ , donde  $M$  es el número de puntos de vista.
- **cameraMatrix1 ( IntPtr )** La cámara de entrada / salida matrices  $[fxk \ 0 \ CxK; \ 0 \ fyk \ cyk; \ 0 \ 0 \ 1]$ . Si se especifican **CV\_CALIB\_USE\_INTRINSIC\_GUESS** o **CV\_CALIB\_FIX\_ASPECT\_RATIO**, algunos o todos de los elementos de las matrices debe ser inicializado.
- **distCoeffs1 ( IntPtr )** Los vectores de entrada / salida de coeficientes de distorsión para cada cámara,  $4 \times 1$ ,  $1 \times 4$ ,  $5 \times 1$  o  $1 \times 5$ .
- **cameraMatrix2 ( IntPtr )** La cámara de entrada / salida matrices  $[fxk \ 0 \ CxK; \ 0 \ fyk \ cyk; \ 0 \ 0 \ 1]$ . Si se especifican **CV\_CALIB\_USE\_INTRINSIC\_GUESS** o **CV\_CALIB\_FIX\_ASPECT\_RATIO**, algunos o todos de los elementos de las matrices debe ser inicializado.
- **distCoeffs2 ( IntPtr )** Los vectores de entrada / salida de coeficientes de distorsión para cada cámara,  $4 \times 1$ ,  $1 \times 4$ ,  $5 \times 1$  o  $1 \times 5$ .
- **ImageSize ( MCvSize )** Tamaño de la imagen, que se utiliza sólo para inicializar la matriz de cámara intrínseca
- **R ( IntPtr )** La matriz de rotación entre la primera y la segunda cámaras de sistemas 'de coordenadas.
- **T ( IntPtr )** El vector de traslación entre las cámaras de los sistemas 'de coordenadas.
- **E ( IntPtr )** La matriz de salida esencial opcional.
- **F ( IntPtr )** La matriz fundamental de salida opcional.
- **termCrit ( MCvTermCriteria )** Criterios de terminación para el algoritmo iterativo optimization.
- **banderas ( CALIB\_TYPE )**.

Las estimaciones de transformación entre las 2 cámaras haciendo un par estéreo. Si tenemos una cámara estéreo, donde se fija la posición relativa y orientatation de las 2 cámaras, y si calculamos poses de un objeto con respecto a la cámara de puño y con la segunda cámara,  $(R1, T1)$  y  $(R2, T2)$  , respectivamente (que se puede hacer con `cvFindExtrinsicCameraParams2`), obviamente, esas posturas se relacionan entre sí, es decir, da  $(R1, T1)$ , debería ser posible calcular  $(R2, T2)$  - sólo tenemos que conocer la posición y la orientación de la cámara segunda con relación a la primera cámara. Eso es lo que hace la función descrita. Se calcula  $(R, T)$  tal que:  $R2 = R * R1, T2 = T1 + R * T$ .

Se va exponer alguno de los elementos que comprenden a la función anterior. Tiene por salida para este proyecto la `_M1, _M2, _D1, _D2, _R, _T` que son las matrices de proyección y distorsión para cada cámara, al igual que la rotación y translación de una cámara con respecto a la otra, como entrada los vectores 2D `_puntosImagen1` y `_puntosImagen2` que son los puntos adquiridos de cada plano imagen con la función `CVFINDCHESSBOARDCORNERS()` y también un vector 3D que guarda los puntos 3D del objeto como lo es para este caso, las distancias entre las esquinas del damero. El termino `cvTermCriteria` ejecuta el proceso de calibración en una forma iterativa para hallar los parámetros anteriores.

Durante este proyecto se ha aplicado la matriz de los parámetros intrínsecos y de distorsión de cada cámara como parámetros iniciales. Por ello se realizó la calibración individual de cada cámara en el capítulo anterior, logrando así mejores resultados. `CV_CALIB_ZERO_TANGENT_DIST` indica que la distorsión tangencial es despreciable o inexistente, y por tanto no es necesario calcular los parámetros de distorsión debidos a tal efecto. `CV_CALIB_FIX_ASPECT_RATIO` hace que los píxeles de la cámara se consideren cuadrados durante la calibración y por tanto las focales `fx` y `fy` sean iguales para cada cámara.

**2.7.2 Corrección de la distorsión y rectificación de las imágenes.** En este punto se utilizan los parámetros de salida de la función `cvStereoCalibrate()`, para incluirlos en la función de rectificación que calcula los parámetros de proyección, rotación y reproyección.

Para la rectificación de las imágenes se aplicó el algoritmo de BOUGUET, que es muy fácil de ejecutar, gracias a que hace parte de las librerías de OpenCV, arroja datos bastante precisos, minimiza las distorsiones introducidas durante la rectificación y así implementa una de sus grandes características que es aumentar al máximo el área de visión cuando se hayan transformado las imágenes.

La función que aplica el algoritmo de BOUGUET es:

```
cvStereoRectify (&_M1, &_M2, &_D1, &_D2, imageSize, &_R, &_T, &_R1, &_R2,
&_P1, &_P2, Q, 0);
```

Ya se conocen las matrices intrínsecas y de distorsión  $_M1$ ,  $_M2$ ,  $_D1$ ,  $_D2$ , al igual que la rotación  $_R$  y translación  $_T$ , ahora se tienen como nuevos datos.

- $_R1$  es la rotación de la cámara izquierda
- $_R2$  es la rotación de la cámara derecha

Esto resulta porque al rotar las dos cámaras para buscar un plano en común medio, se reduce el error a la mitad en cuestión de rotación.

- $_P1$  matriz proyección de la cámara izquierda
- $_P2$  matriz proyección de la cámara derecha

Estas proyecciones actúan de forma independiente.

- $_Q$  matriz de reproyección

La matriz  $_Q$  vincula a las dos cámaras y permite proyectar puntos desde una imagen al mundo real.

$$Q = \begin{bmatrix} 1 & 0 & 0 & -C_x \\ 0 & 1 & 0 & -C_y \\ 0 & 0 & 0 & f \\ 0 & 0 & -1/T_x & (C_x - C'_x)/T_x \end{bmatrix}$$

Donde,  $C_x$  y  $C_y$  son el punto principal en píxeles de la cámara izquierda,  $C'_x$  es el punto principal en la coordenada  $X$  de la cámara derecha,  $T_x$  es la translación en eje  $X$  (en centímetros) y  $f$  es la distancia focal de las dos cámaras.

Comandos para hallar los mapas de píxeles que rectificaran las imágenes.  
Se debe aplicar una por cada imagen.

```
CVINITUNDISTORTRECTIFYMAP (&_M1, &_D1, &_R1, &_P1, mx1, my1);
CVINITUNDISTORTRECTIFYMAP (&_M2, &_D2, &_R2, &_P2, mx2, my2);
```

Como los mapas de píxeles y la matriz  $Q$  se guardan para un posterior uso en la rectificación de imágenes y calculo en las coordenadas 3D de un objeto, se debe tener en cuenta que el uso de estos, se realiza hasta que haya algún movimiento de una cámara con respecto a la otra y en ese caso se tiene que volver a calibrar.

**2.7.3. Resultados de la calibración.** Después de conseguir la matriz  $Q$ , para obtener los puntos reales de un objeto a partir de imágenes bidimensionales, se

guarda en un archivo .XML con la función `cvSave` para un uso posterior en el procesamiento de imágenes. El resultado de la calibración será evaluado en la visualización de las imágenes rectificadas por el usuario de la calibración.

Con las siguientes sentencias se aplican las correcciones a las imágenes, quedan listas para ser visualizadas y utilizadas en el cálculo de la obtención de los puntos.

```
cvRemap (img1, g_img1r, mx1, my1);  
cvRemap (img2, g_img2r, mx2, my2);
```

De tal manera, se obtendrán dos imágenes rectificadas y libres de distorsión.

El usuario debe tener en cuenta que para registrar una óptima calibración, un punto del objeto proyectado en una de las imágenes, debe estar en la misma línea horizontal de la otra o a la misma altura en píxeles, convirtiendo así la búsqueda de puntos en un problema unidimensional o de búsqueda en el eje  $x$  de la imagen como se muestra en la Figura 17.

En caso diferente, se debe realizar nuevamente la calibración.

Figura 17. Rectificación de imágenes



### **3. SEGUIMIENTO ESTEREO**

Este capítulo comprende la parte principal del proyecto, el cual hace uso de parámetros obtenidos en la calibración. Se registra el salto y se proyecta el resultado de su ejecución de dos maneras diferentes (visual y estadística). Ahora bien, antes de entrar en el tema principal conviene conocer el escenario en donde se ejecutarán salto y marcadores.

#### **3.1. ESCENARIO DE CAPTURA**

Es el espacio donde se realizan las pruebas de salto vertical. En este proyecto se aplicará el salto Squat Jump y Counter Movement Jump (contra movimiento). El escenario de trabajo debe tener mínimo 3.0 metros de altura, dimensión suficiente para registrar el marcador del centro de gravedad (CG).

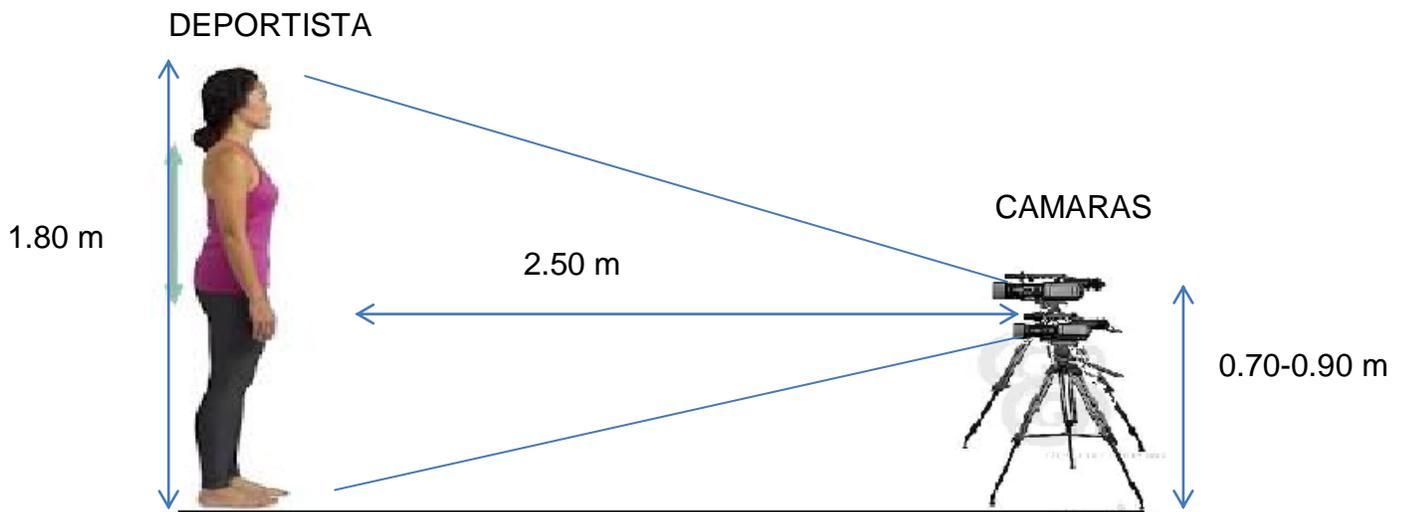
El equipo de cómputo y las cámaras se posicionarán a una distancia de 2.50 metros del deportista y a una altura entre 70 a 90 cm del piso. Los centros ópticos deben apuntar de forma que sus ejes ópticos queden lo más paralelo posible al piso y deben estar a una distancia entre 25 a 40 cm, los parámetros sugeridos anteriormente es para que las cámaras alcancen un mayor campo de visión donde el deportista realiza el salto y no haya pérdida de ningún marcador por lejanía o por salirse del espacio capturado. Ignorar estas recomendaciones, provocará una mayor complejidad en la relación de las cámaras cuando se calibren, en el momento del salto y ocasionarán errores en la obtención de datos.

Otro factor a tener en cuenta en el escenario es la iluminación del mismo. Se aconseja que la luz no incida de forma directa sobre el escenario, que la fuente de luz esté a espaldas de las cámaras y que recree una luz de manera suficientemente difusa para que las cámaras reconozcan los marcadores.

En este proyecto, los autores trabajaron en un escenario con una mezcla entre luz natural y luz de bombillos fluorescentes de color blanco. La luz fue suficiente y ajustada para una óptima adquisición de los puntos de los marcadores.

En la Figura 18 se muestran las dimensiones del escenario para el uso de este proyecto.

Figura 18. Escenario de captura



### 3.2. EL SALTO<sup>9</sup>

Según Jaramillo (1999), la saltabilidad es la capacidad de manifestar de forma explosiva el esfuerzo muscular para realizar una acción efectiva sin apoyo en el aire. En otras palabras, es una cualidad compleja compuesta por fuerza, velocidad y habilidad. Así mismo, el salto es una actividad física que se caracteriza por esfuerzos musculares cortos de carácter “explosivo” que tiene muchos estilos, cuya técnica adquiere primordial importancia (A.V Postoev, 1991).

Las Fases de Categorización del Movimiento del Salto son:

- **Fase de Preparación:** como característica del movimiento de caída del centro de masa, tiene las siguientes subfases:
  - ✓ *Equilibrio:* se presenta únicamente en la caída de una serie de saltos de rebote. Como característica, la velocidad de caída es la misma al comienzo y al final de la fase.
  - ✓ *Compresión:* el final de la fase está determinado por la velocidad en el punto más bajo del centro de masa.
- **Fase de Propulsión:** caracteriza el impulso hacia arriba del centro de masa y tiene las siguientes subfases:
  - ✓ *Aceleración:* la característica es que la velocidad y la aceleración en el levantamiento son positivas.
  - ✓ *Desaceleración:* la característica es que la velocidad es positiva y la aceleración negativa.

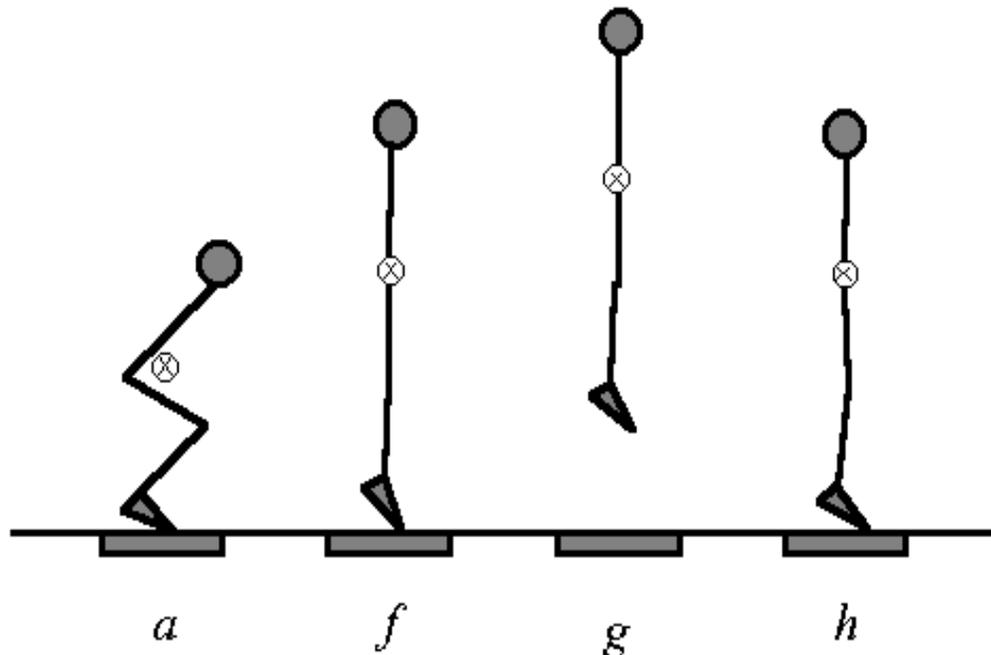
---

<sup>9</sup>Bosco, C. “La valoración de la fuerza con el test de Bosco” Editorial Paidotribo. Barcelona.1994

En el presente trabajo se aplicaron dos tipos de saltos los cuales hacen parte de las pruebas estandarizadas; a saber:

- **Squat Jump**, o salto desde posición de semisentadilla con las rodillas flexionadas a 90°. En la Figura 19 se puede observar la secuencia del salto squat jump. En el paso **a** se flexionan las piernas hasta conseguir un ángulo de 90° y luego se espera hasta el paso **f** para realizar el despegue.

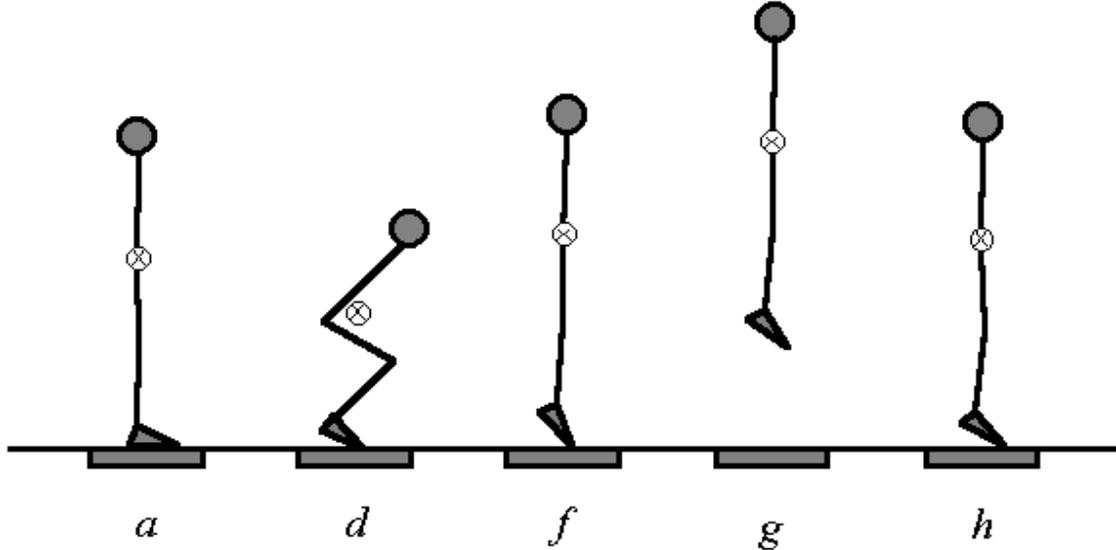
Figura 19. Secuencia de salto Squat Jump



Fuente: <http://www.brunel.ac.uk/~spstnpl/BiomechanicsAthletics/CMJSJ.gif>

- **Counter movement Jump**, o salto desde posición de pies con acción de contra movimiento (Figura 20). En el paso **a** se observa que el sujeto se encuentra en reposo y erguido, seguido en tres tiempos de una sentadilla (paso **d**) que lo impulsa al momento del despegue como se muestra en el paso **f**.

Figura 20. Secuencia de salto Countermovement Jump



Fuente: <http://www.brunel.ac.uk/~spstnpl/BiomechanicsAthletics/CMJSJ.gif>

### 3.3. MARCADORES

Dada su importancia para el proyecto, fue necesario utilizar dos tamaños y colores diferentes hasta encontrar el marcador apropiado. Inicialmente se intentó con pelotas blancas de icopor de tamaño menor al de una pelota de ping pong. Sin embargo, para encontrar los marcadores había que contrastar claramente, las pelotas blancas con el traje negro del deportista y el fondo oscuro del escenario visualizado por la cámara. Este primer intento presentó diferentes dificultades. Por tal razón fue necesario pintar las pelotas de color verde fosforescente y aumentar su tamaño a un diámetro igual al de las pelotas de ping pong. En algunas ocasiones conviene que el deportista vista prendas bastante ajustadas de color negro específicamente en las piernas, para lograr un mayor resalte de la pelota. Gracias a los dos cambios introducidos por los autores, se alcanzó una mayor eficiencia en el montaje del escenario y captura de los marcadores.

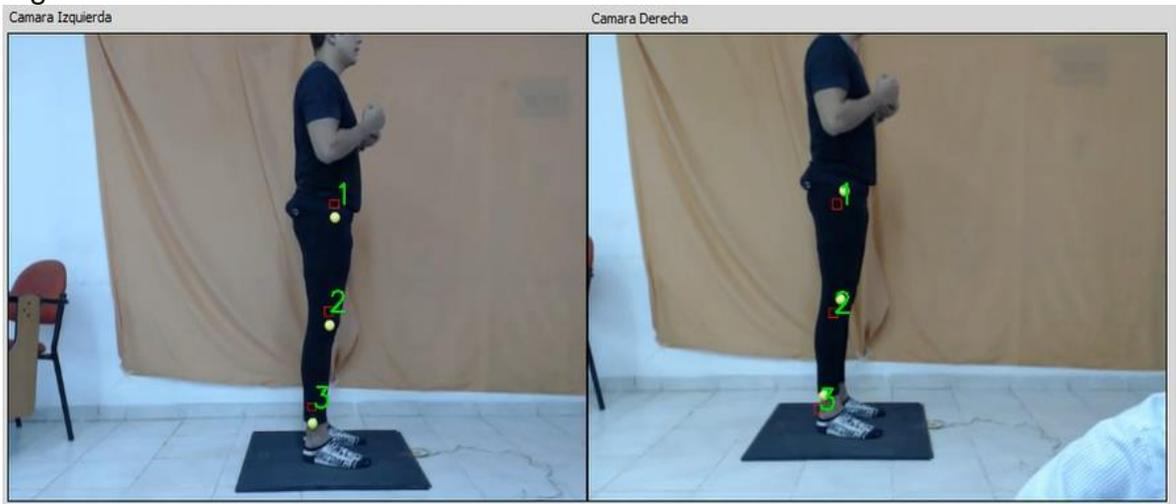
Los colores fosforescentes son poco comunes en los ambientes deportivos. Por esta razón, constituyen un marcador móvil y ligero debido al peso del icopor. Ver Figura 21.

Figura 21. Marcadores



**3.3.1 Ubicación de los marcadores<sup>10</sup>.** Los marcadores se establecen de una forma jerárquica en la anatomía del sujeto a analizar. Se colocan en puntos representativos para la descripción del movimiento del salto. En el proyecto se ubican tres marcadores en el lado derecho de la extremidad inferior del deportista: El primero y de mayor altura se encuentra ubicado en la parte media entre la espina iliaca y cabeza femoral. Este punto se encuentra bastante cerca al centro de gravedad y se representará como (CG); el segundo, se localiza en el Epicóndilo femoral lateral o en la parte lateral de la rodilla y el tercero se ubica en la parte superior del maléolo lateral o comúnmente llamado tobillo. Ver Figura 22.

Figura 22. Ubicación marcadores



<sup>10</sup>Diseño e implementación de un sistema de videogrametría digital para el análisis de saltabilidad en deportistas. (2010)

### 3.4. DESARROLLO DEL SEGUIMIENTO ESTEREO Y EL ALGORITMO

Al emplear un sistema estéreo se requiere trabajar de forma sincronizada; con ello se pretende evitar errores de desfase al momento de capturar el movimiento a través de las cámaras. Prescindir de este método conduciría al siguiente problema:

La cámara número uno obtiene su imagen con un determinado tiempo de exposición. Capta el movimiento del deportista en cierto espacio; posteriormente la cámara número dos captura la imagen después de un desfase de tiempo con respecto a la cámara número uno, lo cual origina una vista desplazada del movimiento por tratarse de un salto continuo y rápido.

Para solucionar el problema se trabaja con **multihilos (MultiThreads)**, estos son subprocesos que pueden ejecutarse individualmente y en forma paralela o sincronizada.

**3.4.1. Trabajo en paralelo de las cámaras y el algoritmo.** Hay dos maneras de trabajar la sincronización o paralelismo: tipo hardware y tipo software.

- Tipo Hardware: requiere de un complejo circuito electrónico de sincronización cuya eficiencia es bastante amplia.
- Tipo Software: no requiere de recursos externos, solo de una buena programación sincronizada, pues, de esta manera se evitan gastos extras. Para este proyecto se recomienda utilizar este tipo de sincronización considerando que produce resultados muy similares a los obtenidos por el tipo hardware. Se sugiere emplear hardware externo cuando se trabaje con más de 3 cámaras.

El método de sincronización no solo queda empleado en la captura de las imágenes, también incluye el desarrollo del procesamiento de los marcadores y obtención de los puntos. Lo anterior se hace con el propósito de alcanzar mayor número de **frames** o imágenes por segundo (FPS).

Se crean dos hilos de ejecución (**Threads**) con las librerías de Qt Qthread. Cada uno de ellos crea a su vez un subproceso encargado de ejecutar los apartados de rectificación, el procesamiento de los marcadores, la adquisición y filtro de los puntos y el seguimiento de ellos en cada imagen. Todo esto se realiza durante el proceso principal del programa y estos comienzan a trabajar en paralelo.

**3.4.1.1. Rectificación de las imágenes.** Durante esta etapa, se corrigen las imágenes a partir de los mapas de pixeles encontrados en la calibración estéreo, con el fin de facilitar la obtención de los puntos proyectados. De esta manera queda establecida la búsqueda de correspondencias en una sola dimensión. En

otras palabras, un punto encontrado a una altura en una de las dos imágenes, se localizará a la misma altura de la otra imagen con una diferencia en el eje **x**. Para ejecutar la rectificación se debe hacer lo siguiente:

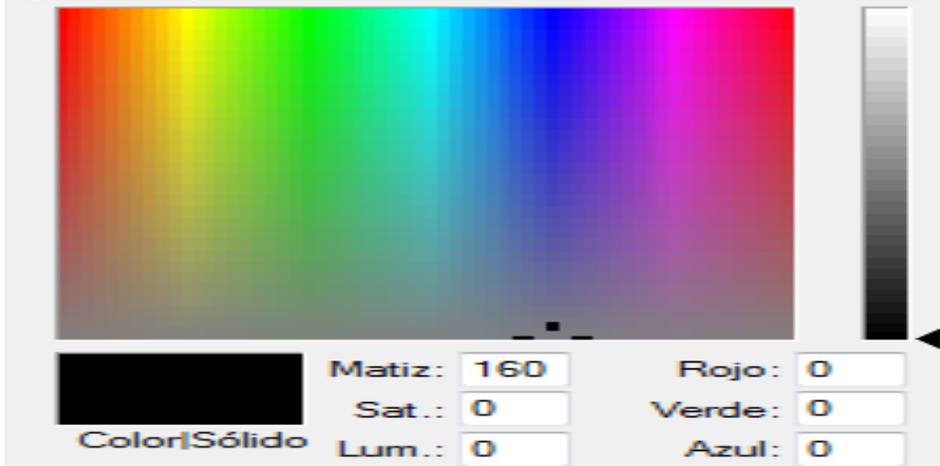
- Cargar los mapas de pixeles *mx* y *my* correspondientes a cada cámara. Con la función *cvLoad()* se puede obtener los parámetros guardados.
- Rectificar las imágenes incluyendo los mapas de pixeles en la función *cvRemap()*.

**3.4.1.2. Procesamiento de los marcadores.** En este apartado se explica cómo se extraen los marcadores de las imágenes adquiridas por la cámara mediante el procesamiento digital de imágenes, con el fin de obtener los puntos bidimensionales verídicos de los marcadores. Este apartado se divide en tres procedimientos importantes:

- Segmentación: es el proceso de dividir una imagen digital en varias partes (grupos de pixeles) o bloques. El objetivo principal de la segmentación es simplificar y/o cambiar la representación de una imagen en otra más significativa y más fácil de analizar. La segmentación se usa para representar con una etiqueta a cada píxel de la imagen de forma que los píxeles que compartan la misma etiqueta también tengan ciertas características visuales similares.

El proceso de segmentación se aplica transformando la imagen del modelo RGB al HSV con la función *cvCvtColor()*, luego se procede a trabajar con la capa H (Hue-matiz), que representa el color principal o estado puro del color. Por último se convierte la imagen **H** en una imagen a escala de grises o blanco y negro por medio de la umbralización. En ese momento es importante determinar un buen rango del color, pues, escoger un valor acertado implica una segmentación de muy buena calidad y la fácil obtención del bloque de pixeles del marcador. Para obtener el mejor rango de color, conviene usar visualmente una imagen que muestre el espacio de colores. Ver Figura 23.

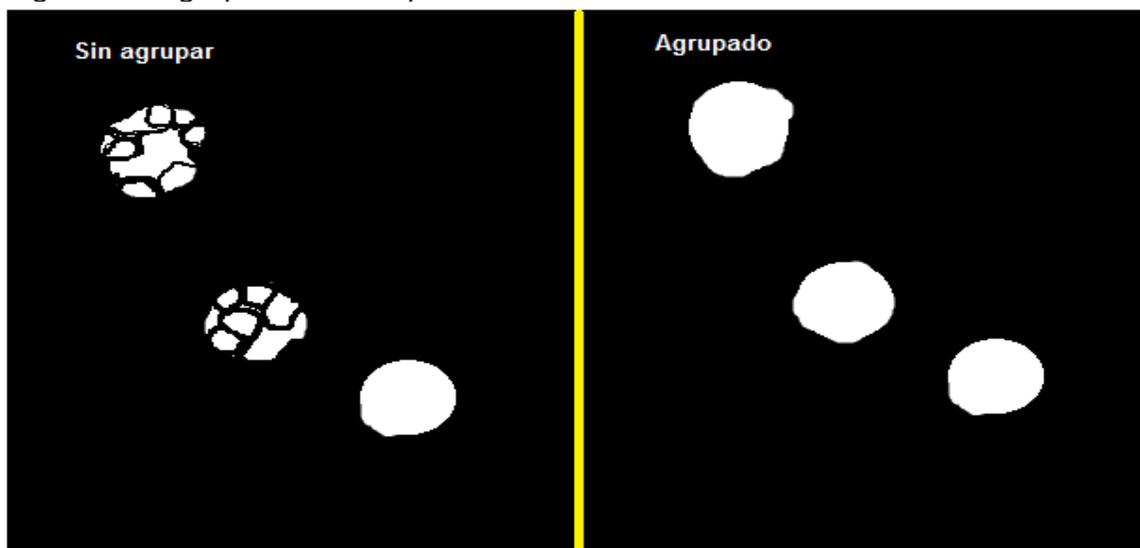
Figura 23. Espacio de Colores



En el proyecto se aplicó la umbralización con la función *cvThreshold()* con un rango de valores de 28-50 (este rango se puede graduar en el programa) la cual se refiere al color verde fosforescente que se utilizó en los marcadores. Gracias a la umbralización se puede quitar el ruido que presenta el escenario.

- Agrupamiento o consolidación de regiones: los marcadores representan un grupo sólido de píxeles en la imagen. No obstante, debido a alteraciones en la luminosidad y a otros factores, al realizar la operación de segmentación puede suceder que un marcador aparezca representado en muchas regiones diferentes y desconectadas. La función *cvMorphologyEx()* las consolida en una sola región o grupo que identifica un marcador. Ver Figura 24.

Figura 24. Agrupamiento de píxeles

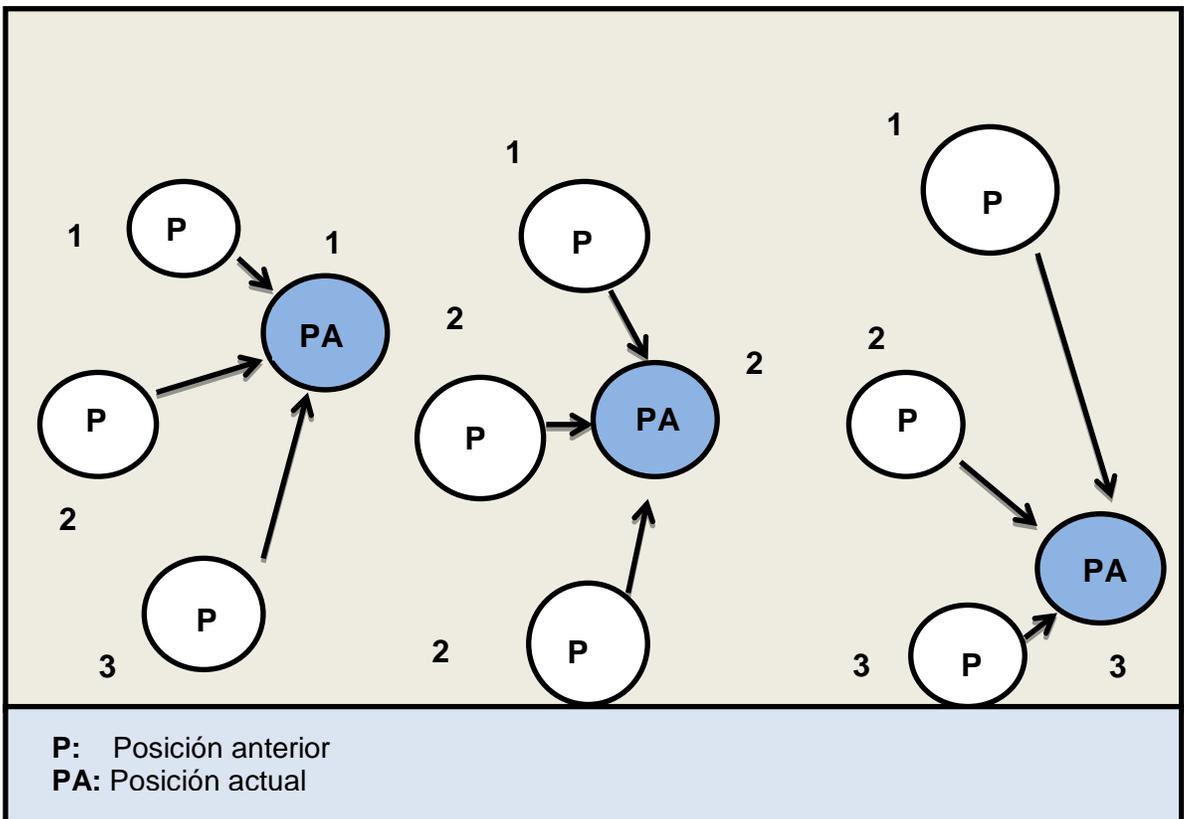


- Eliminación de falsos positivos. En el proyecto se optó por eliminar regiones de píxeles no referentes a los marcadores con un filtro determinado por el área de píxeles de cada región. El comando `cvFilterByArea()` permite realizar lo anterior. Esta función pertenece a las librerías de `cvBlobs`, encargadas de obtener y etiquetar los bloques de píxeles o marcadores.

**3.4.1.3. Seguimiento de los puntos.** En esta parte del proyecto se sigue la posición de los marcadores después de haber sido detectados y etiquetados en orden jerárquico. Aunque halla ausencia de ellos por unos pocos **frames**, lo ideal es que nunca se pierdan de las imágenes. En casos reales, durante la ejecución del salto puede suceder que algunos queden ocultos debajo de los brazos o que no sean detectados debido a un cambio repentino de luz, el cual modificó sus características visuales ante la cámara. El seguimiento trabaja con las etiquetas que se le han establecido al momento de la detección y lo hace de forma preventiva y predictiva.

- Preventiva: se trata de un patrón muy sencillo, el punto que esté más cerca a ése pertenece. Eso significa que la posición del marcador detectado pertenece a aquel que esté más cerca de él, con relación a la posición de la anterior imagen. Ver Figura 25.

Figura 25. Seguimiento preventivo



- Predictiva: todo movimiento tiene una dirección, sentido y magnitud que se conoce como vector del movimiento. Para encontrar la posición actual de un marcador no detectado, se calcula el vector generado por las posiciones anteriores a la actual y a las dos anteriores. Una vez calculado este vector, se suma a la primera de las dos posiciones anteriores y así se halla la posición actual del marcador. La magnitud del vector calculado para encontrar la nueva posición, se divide por el número de **frames** o imágenes en donde se encuentre perdido el marcador.

En el programa la función que ejecuta el seguimiento se llama *Trackingg()*.

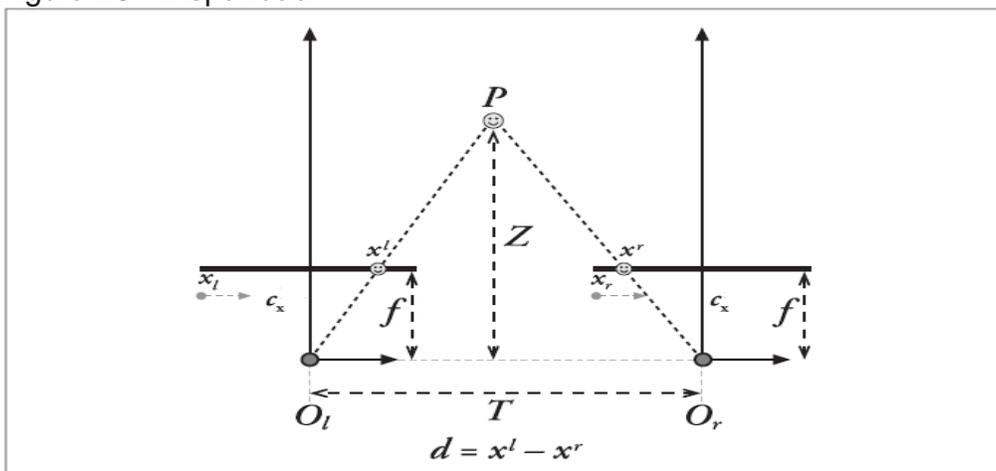
**3.4.2. Estimación de las distancias reales.** Es la parte que describe el comportamiento de los marcadores, al encontrar las posiciones reales (**X, Y, Z**) con respecto al eje óptico de la cámara izquierda.

**3.4.2.1 Disparidad<sup>11</sup>.** La disparidad se puede concebir para el eje **x** o **y**. En este proyecto se realiza en el eje **x**.

$$d = X^l - X^r \quad \text{Ecuación 13}$$

Donde  $X^l$  y  $X^r$  son las coordenadas en el eje **x**, que se refieren a las cámaras izquierda y derecha respectivamente. Las estimaciones de la distancia inician con la disparidad que existe de un objeto proyectado en cada cámara. La disparidad **d** entonces se define como la diferencia en píxeles entre las proyecciones en las imágenes izquierda y derecha debidas a un mismo punto 3D del mundo real. Ver Figura 26.

Figura 26. Disparidad



Fuente: O'Reilly Learning OpenCV, Computer Vision with the OpenCV Library

<sup>11</sup>Gary Bradski and Adrian Kaehler. Computer vision with the OpenCV library. O'Reilly. (2008)

Se debe tener en cuenta que la disparidad se aplica porque llegado a este punto, las imágenes deben estar rectificadas y los marcadores correctamente identificados, asociados y etiquetados en cada imagen.

$$Z = \frac{fT}{X^l - X^r} \quad \text{Ecuación 14}$$

La Ecuación 14, es la responsable de obtener la distancia de profundidad real de un marcador, se deduce a partir de la relación de semejanza de triángulos existente en la Figura 26.

Para hallar las dos dimensiones restantes  $X$  y  $Y$ , se despejan de las Ecuaciones 5 y 6 del Capítulo 1 respectivamente. En cada una se debe incluir el desplazamiento del punto principal según sea su eje ( $C_x$ ,  $C_y$ ).

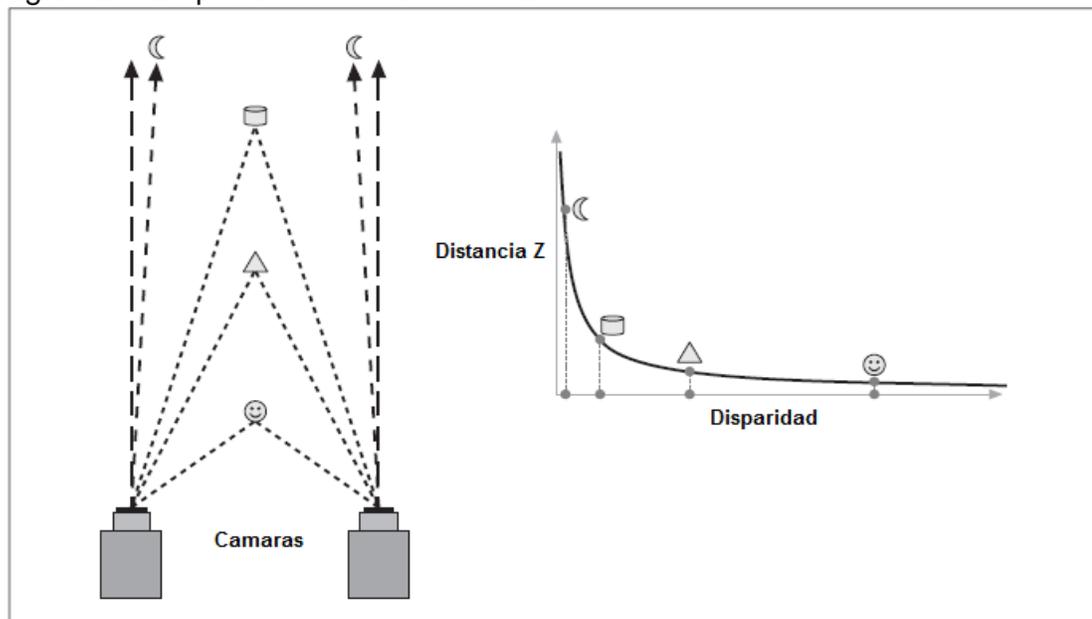
Entonces las ecuaciones resultantes son las siguientes,

$$X = (x - C_x) * Z/f \quad \text{Ecuación 15}$$

$$Y = (y - C_y) * Z/f \quad \text{Ecuación 16}$$

La Figura 27 muestra la relación existente entre la disparidad y distancia  $Z$  de un objeto real, se ve como son inversamente proporcional.

Figura 27. Disparidad vs la distancia z



Fuente: O'Reilly Learning OpenCV, Computer Vision with the OpenCV Library

**3.4.3. Grabación del salto.** Antes de la grabación del salto se debe calibrar la posición de las cámaras como un solo objeto (no se pueden mover individualmente) para establecer que el eje óptico quede paralelo al piso. El procedimiento es utilizar una tabla con tres marcadores adheridos a diferentes alturas pero en el mismo plano; luego posicionar la tabla de manera perpendicular al piso y medir las distancias de profundidad a que se encuentran los marcadores a las cámaras. Esto se realiza en el programa con el fin de imprimir los valores de las distancias. Si esas distancias son iguales para los tres casos, quiere decir que el eje óptico de las cámaras se encuentra completamente paralelo al piso. La razón de la calibración es para referenciar un plano perpendicular al eje óptico que ayudará a encontrar valores bastante precisos de la altura del deportista. Ver Figura 28.

Figura 28. Tabla para la calibración



La etapa de grabación tendrá 80 **frames** o imágenes, equivalentes aproximadamente a 4 segundos de captura y suficientes para efectuar el salto. Su salida son las posiciones 3D real de todos los marcadores involucrados. El archivo de salida es de extensión .XML para uso posterior en los resultados estadísticos, así como para la visualización 3D en tiempo control.

Figura 29. Diagrama de flujo del seguimiento estéreo

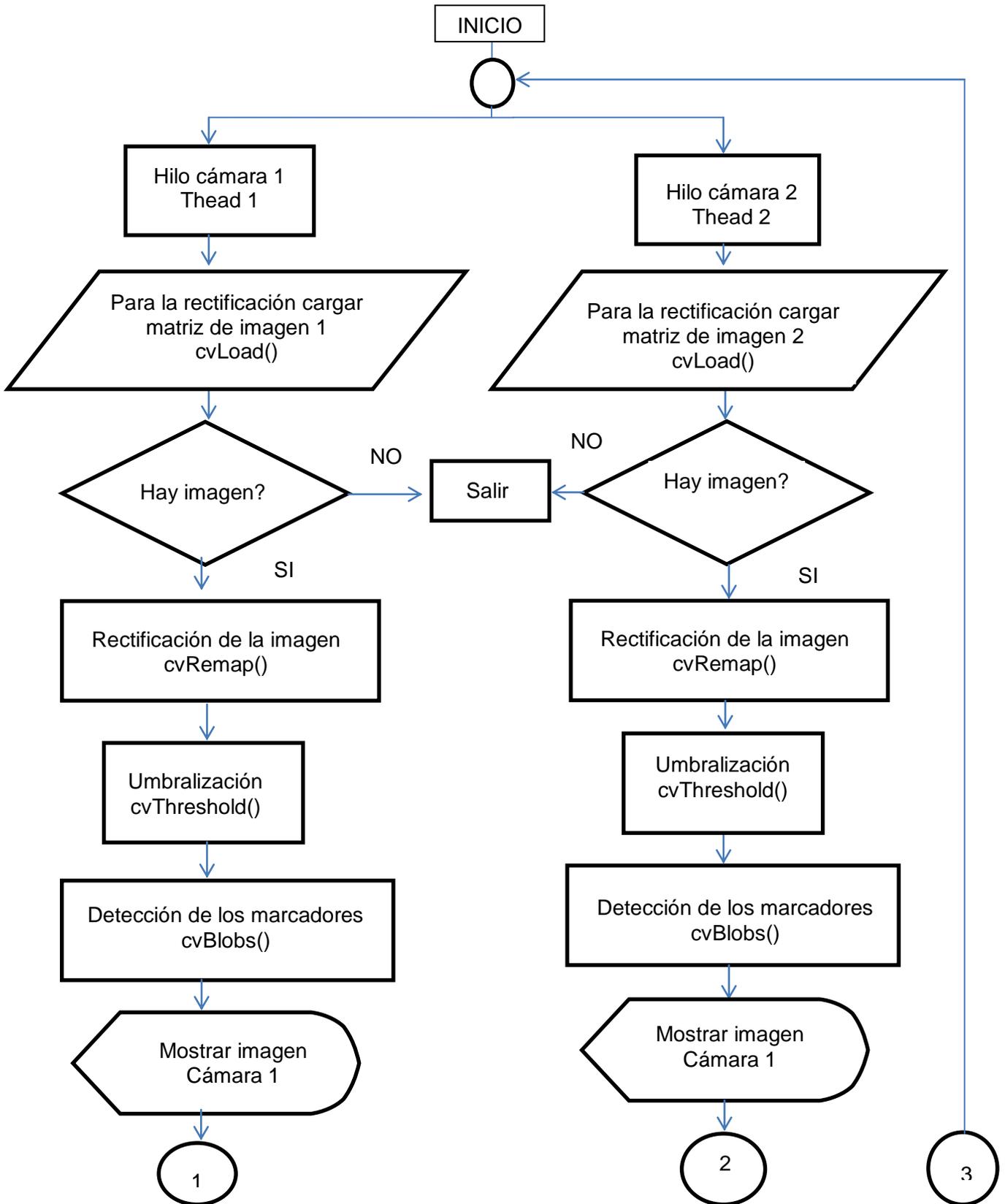
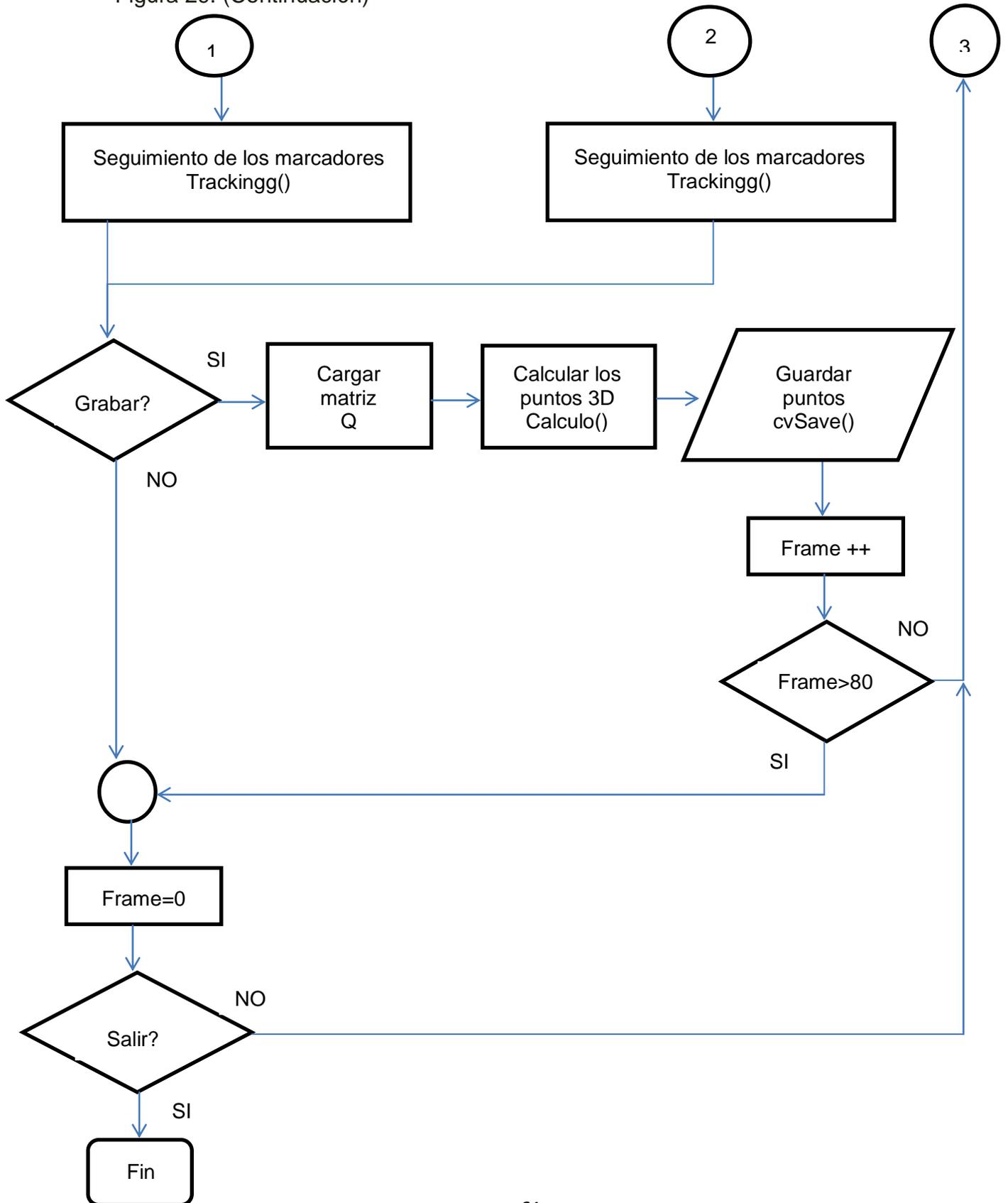


Figura 29. (Continuación)



## 4. VISORES 3D

La visualización del movimiento queda a cargo de los visores 3D, los cuales presentan un modelo biomecánico de la extremidad inferior (pierna), que va desde la cadera al tobillo. Hay dos tipos de visores:

### 4.1. VISOR 3D TIEMPO REAL

Es una de las partes más entretenidas del programa. Al realizarse un movimiento humano como el salto, de manera virtual y simultánea, cualquier movimiento capturado por las cámaras puede ser representado inmediatamente en el visor 3D. Ver Figura 30.

El algoritmo ejecuta el mismo procedimiento explicado en el apartado de estimación de distancias para hallar las posiciones 3D de los marcadores y realiza la función de dibujar gracias a las librerías *OpenGL*, que se resume en tres funciones:

- `initializeGL()`: adaptar las luces de la escena.
- `paintGL()`: su principal función es pintar los objetos de la escena, aquí se incluye el modelo biomecánico de forma 3D.
- `resizeGL()`: como se va a ver la escena, su perspectiva y distancia a observar los objetos, es como la cámara del visor.

Figura 30. Visor tiempo real

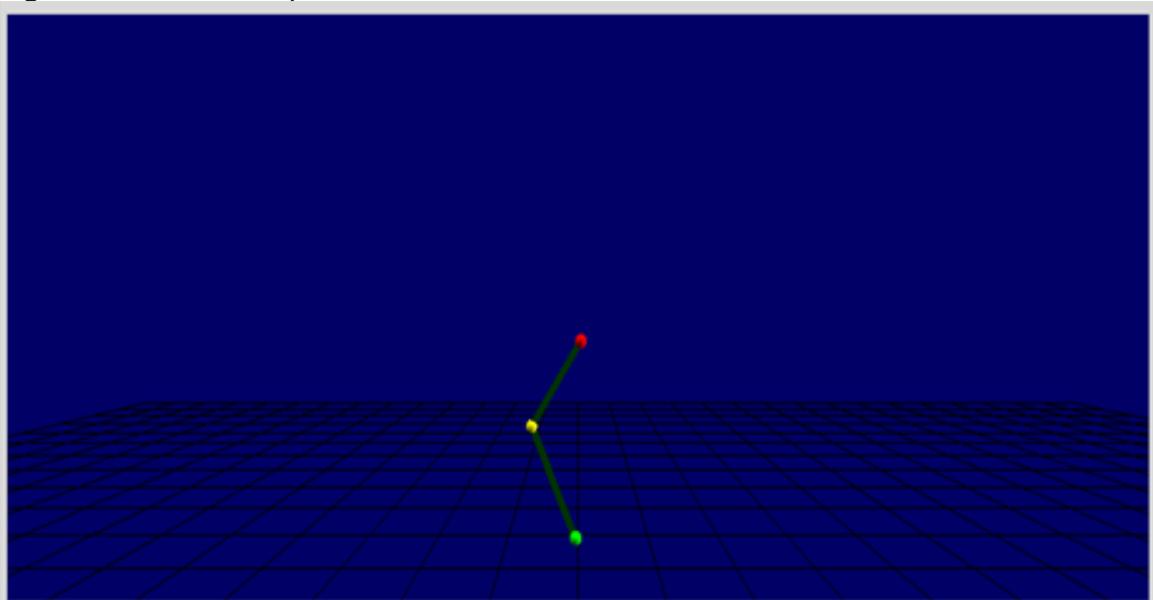
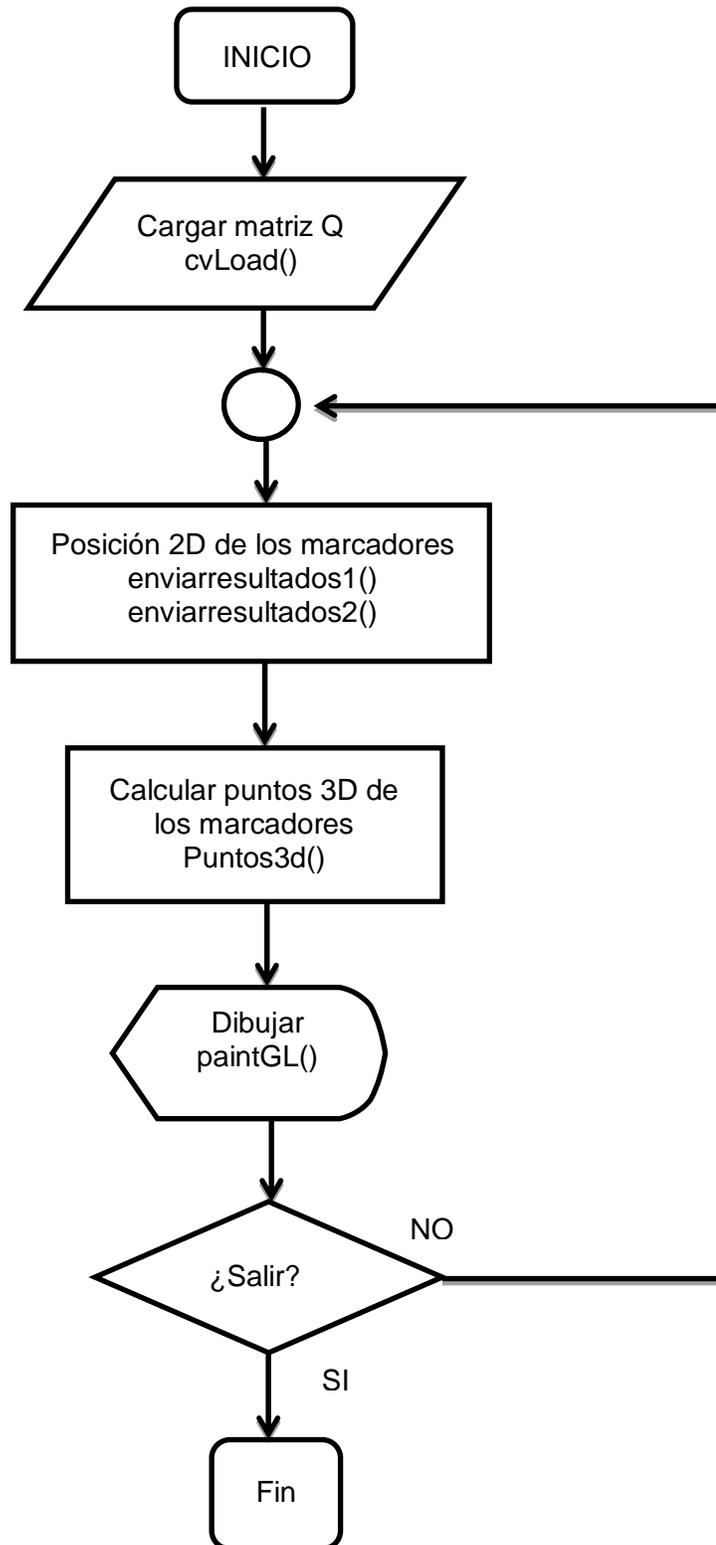


Figura 31. Diagrama de flujo del algoritmo de tiempo real



## 4.2. VISOR 3D TIEMPO CONTROL

El visor tiempo control, presenta la misma visualización que el visor tiempo real, con la diferencia que el primero consta de una línea de tiempo que le da control sobre la velocidad, dirección de muestreo y tiene la opción de ver desde cualquier perspectiva el modelo biomecánico. Además, extrae las posiciones reales de los marcadores del archivo que fue grabado durante el salto. Aproximadamente 4 segundos (tiempo de grabación).

Las funciones ejecutadas en el algoritmo son básicamente las mismas que en el apartado anterior, con la diferencia que el muestreo se hace de forma manual.

En la Figura 32 se puede observar el estilo del visor tiempo control, con un modelo biomecánico.

Figura 32. Visor 3D - Tiempo Control

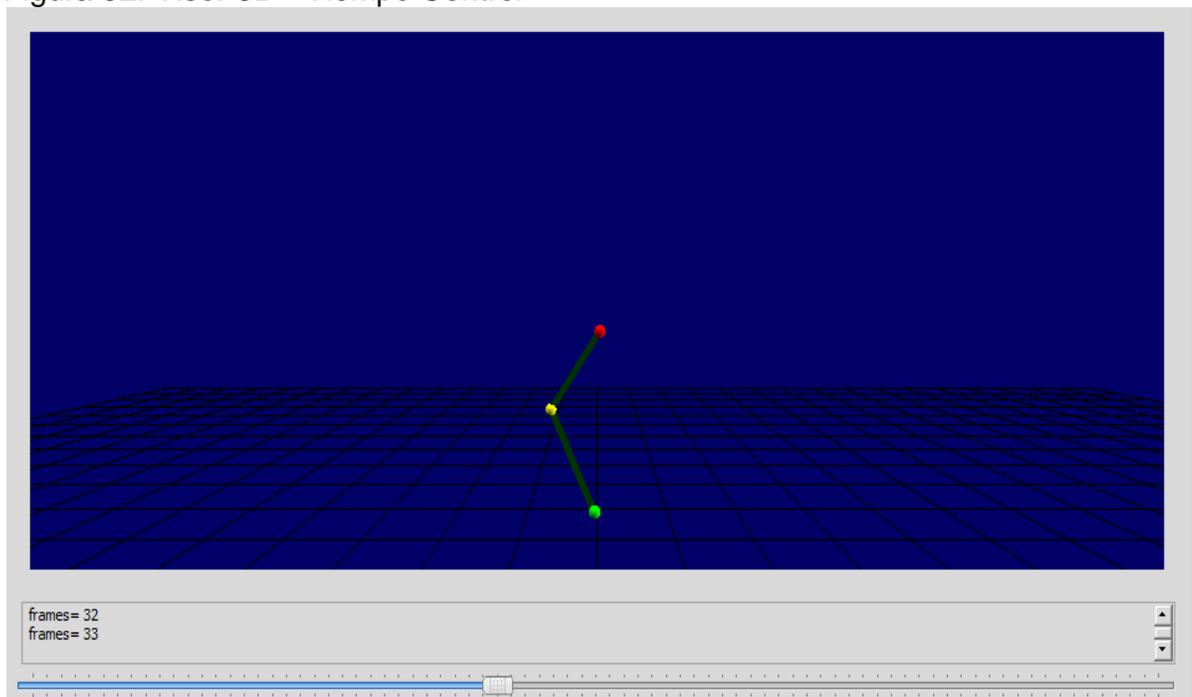
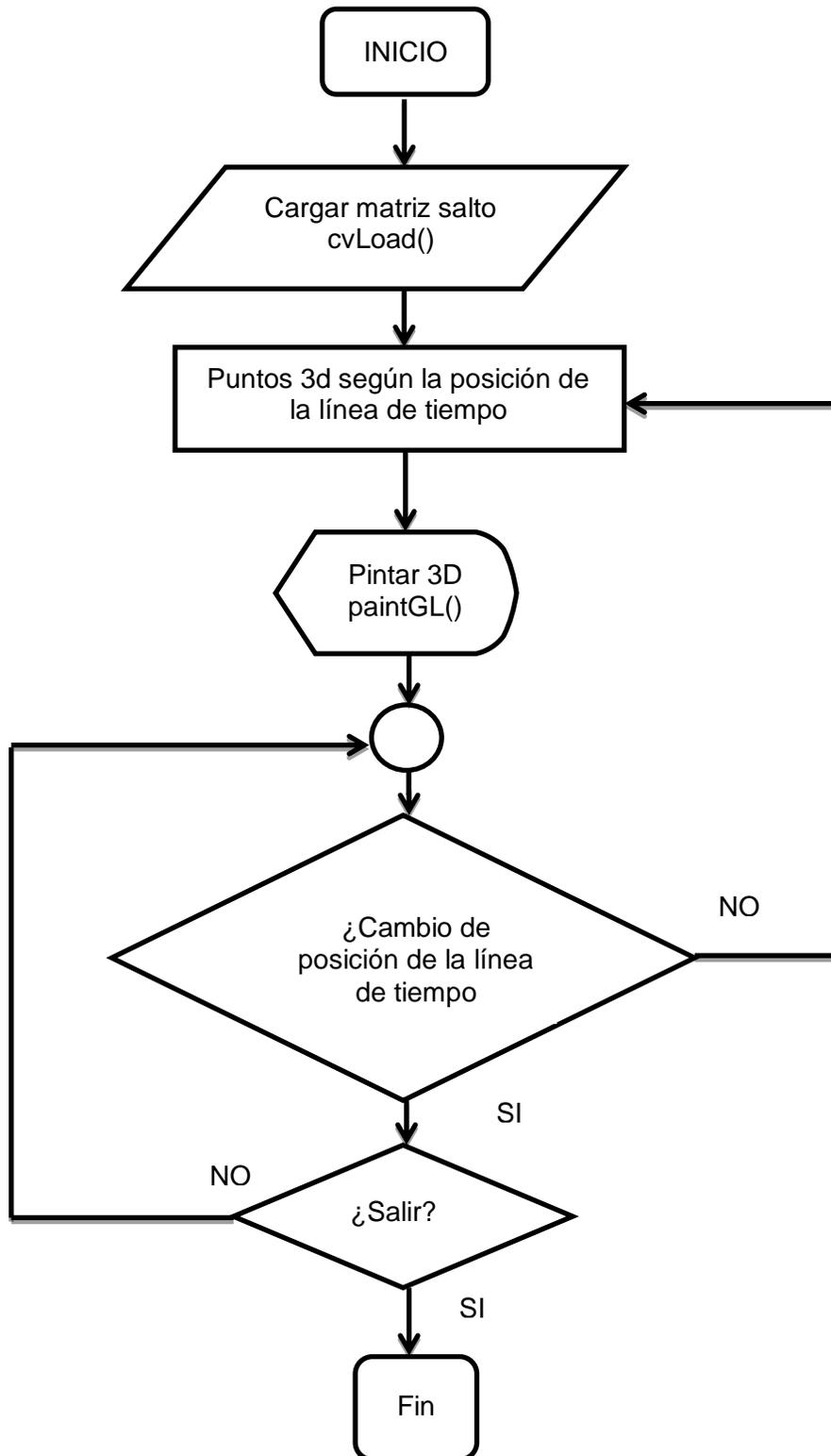


Figura 33. Diagrama de flujo del algoritmo de tiempo control



## 5. RESULTADOS ESTADÍSTICOS

Se muestran gráficamente los resultados de las variables cinemáticas y cinéticas, motivo de análisis para el especialista en saltabilidad.

### 5.1. CÁLCULO DE LAS VARIABLES CINEMÁTICAS Y CINÉTICAS<sup>12</sup>

El método utilizado para el cálculo de las variables, comienza con la carga de la matriz que contiene los datos de la altura del marcador registrado como CG (centro de gravedad). Este archivo fue grabado en la sección de grabación del salto, tuvo una duración de 80 **frames**, ósea, aproximadamente 4 segundos. De igual manera, la representación del tiempo de las gráficas, será de 80 **frames** en el eje **x**. Cada **frame** con una duración de 0,047 segundos resultante de la velocidad de cuadros adquiridos por segundo (FPS) que logró un registro 21.

**5.1.1. Variables cinemáticas.** Las variables cinemáticas del salto son Recorrido del CG (centro de gravedad), Velocidad CG y Aceleración CG.

**5.1.1.1. Recorrido CG (cm).** Es la primera variable a calcular y quizás la más importante, puesto que es la variable inicial y permite hallar las otras que se pueden derivar de ella. La altura del marcador se obtiene de la matriz cargada CG de todos los **frames**; tiene como referencia el primer dato de la altura para disponer de un punto de partida igual a cero. Ver Ecuación 17.

$$\text{Recorrido CG} = \text{Altura CG (Frame 1)} - \text{Altura CG (Resto de frames)}$$

Ecuación 17

**5.1.1.2. Velocidad CG (m/s).** Se determina a partir del recorrido CG, estableciendo en primera instancia una velocidad inicial igual a cero. La diferencia del recorrido será dividida por la duración de cada **frame** que es aproximadamente 0,047 o 1/21 segundos. Como se observa el valor 21, se refiere al número de (FPS) capturados por cada cámara. Ver Ecuación 18.

$$\text{Velocidad CG} = \frac{\text{RecorridoCG(Actual)} - \text{RecorridoCG(Anterior)}}{100 \times (\text{TiempoFrame})} \quad \text{Ec. 18}$$

---

<sup>12</sup>PERDOMO ORTIZ, Diego Fernando. "Diseño e implementación de un sistema de videogrametría digital para el análisis de saltabilidad en deportistas". (2010)

**5.1.1.3. Aceleración CG (m/s<sup>2</sup>).** Está determinada a partir de la velocidad. Por tal razón, se realiza la diferencia de velocidades y luego se divide por el tiempo de duración de un frame. De manera simétrica la velocidad CG se lleva a una aceleración inicial igual a cero. Ver Ecuación 19.

$$AceleraciónCG = \frac{VelCG(Actual) - VelCG(Anterior)}{TiempoFrame} \quad \text{Ec. 19}$$

**5.1.2. VARIABLES CINÉTICAS.** Las variables cinéticas que se trabajaron del salto son La Fuerza Vertical y La Potencia Mecánica.

**5.1.2.1. Fuerza Vertical (Newton).** Para calcular la fuerza vertical se debe usar el peso del deportista; variable que se tiene que incluir en el software al principio del salto. Ver Ecuaciones 20 y 21.

$$PesoDeportista = MasaDeportista(Kg) \times Gravedad(m/s^2) \quad \text{Ec. 20}$$

$$FuerzaVertical = AcelCG \times MasaDeportista(Kg) + PesoDeportista$$

Ecuación 21

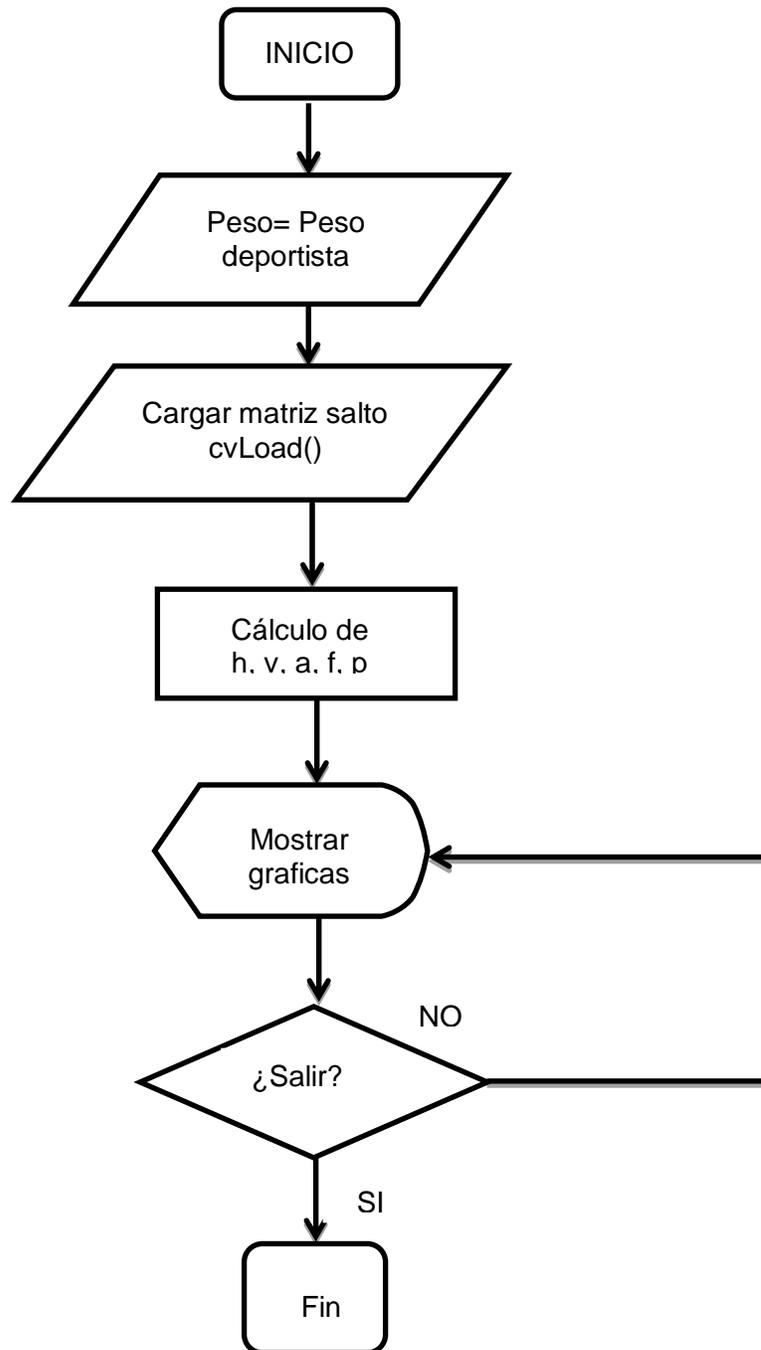
**5.1.2.2. Potencia Mecánica (Watts).** La potencia mecánica se obtiene a partir del producto entre la fuerza vertical y la velocidad CG. Ver Ecuación 22.

$$PotenciaMecanica = FuerzaVertical \times VelocidadCG \quad \text{Ecuación 22}$$

## 5.2. EL ALGORITMO

El programa utiliza librerías de Qwt, con funciones para realizar gráficas estadísticas. Cada variable fue calculada y guardada en un vector a partir de los valores de la altura del marcador CG, los cuales fueron cargados desde el archivo de salto.XML. Los resultados de los cálculos se plasmaron en gráficas sucesivas con la opción de solaparse entre sí.

Figura 34. Diagrama de flujo del algoritmo de resultados estadísticos



## 6. HARDWARE, SOFTWARE Y MATERIALES PARA LA CAPTURA

En este capítulo se detallarán los elementos incluidos en el proyecto:

### 6.1. HARDWARE

Corresponde a todas las partes tangibles de un sistema informático; sus componentes son: eléctricos, electrónicos, electromecánicos, mecánicos y cualquier otro elemento físico involucrado.

**6.1.1. CÁMARAS WEB.** Sistemas de sensores ópticos de forma digital para la adquisición de imágenes y transmisión vía USB a un computador. Son como los ojos de la persona cuando trabajan emparejados. Son capaces de dar una percepción de la profundidad de los objetos. En este proyecto se emplearon dos cámaras web marca Logitech C920 que tienen resolución de vídeo Full HD 1080p y Sensor objetivo Carl Zeiss Full HD, gracias a esto son eficientes para capturar el salto por su velocidad de cuadros por segundo, velocidad de transmisión de información y calidad de captura de la imagen. Ver Figura 35.

Figura 35. Cámaras



**6.1.2. COMPUTADOR (PORTÁTIL O DE ESCRITORIO).** Máquina electrónica capaz de recibir, procesar y devolver resultados en torno a determinados datos y que para realizar esta tarea cuenta con un medio de entrada y uno de salida. Al momento de ejecución y de obtener los datos del salto se usó una computadora portátil con un procesador Intel Core I7 y 6GB de memoria RAM suficiente para procesar el software eficientemente. Ver Figura 36.

Figura 36. Portátil



## 6.2. SOFTWARE

Equipamiento o soporte lógico de un sistema informático. Comprende el conjunto de los componentes lógicos necesarios que hacen posible la realización de tareas específicas, en contraposición a los componentes físicos, llamados hardware.

**6.2.1. Selección del software.** La elección del software a trabajar se realizó teniendo en cuenta la facilidad de uso, los correspondientes conocimientos sobre él y la eficiencia al desarrollar programas.

La plataforma se hizo sobre el programa Qt, aprovechando la amplia gamma de librerías que la comprenden y su diseñador de interfaz. Se incluyeron librerías de *OpenCV* dedicadas al procesamiento de imágenes: También se incorporaron librerías *OpenGL* para la visualización virtual, al igual que *cvBlobs* librerías para la detección de bloques de pixeles de interés y por último, las librerías de Qwt para las gráficas estadísticas.

## 6.3. MATERIALES

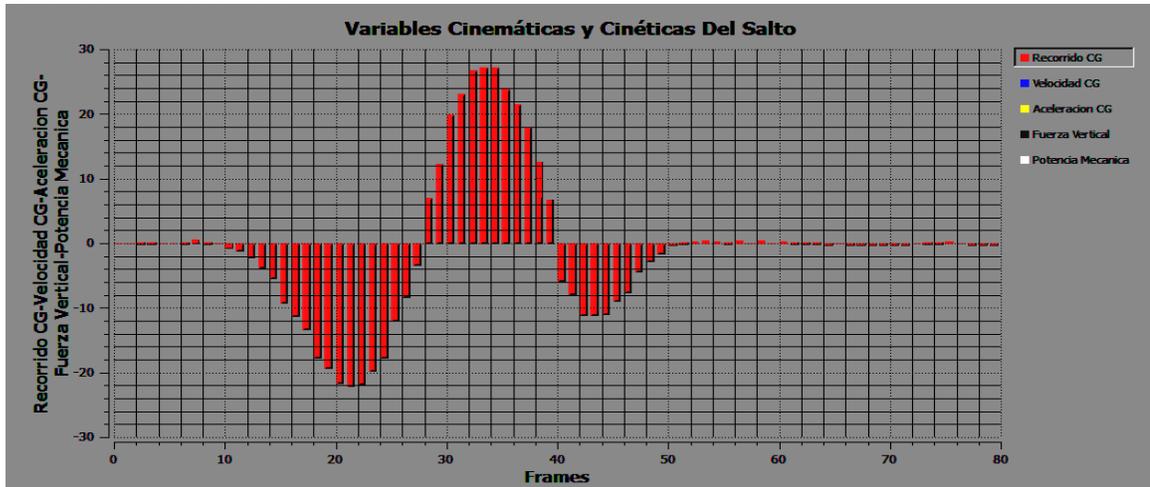
Son los elementos complementarios incluidos en el sistema:

- Marcadores: pelotas verdes fosforescentes de gran resalte.
- Leggins negro: conviene utilizarlos cuando se presentan dificultades al capturar los marcadores.
- Fondo de contraste: fondo que debe contrastar con los colores de los marcadores. Es conveniente utilizarlos cuando se presentan dificultades al capturar los marcadores.
- Damero: tablero de ajedrez impreso y pegado sobre una superficie plana. Utilizado para la calibración de las cámaras.
- Tabla: elemento para posicionar los marcadores al momento de calibrar la cámara izquierda con respecto a la profundidad de cada marcador. De esta manera se logra establecer que el eje óptico de la cámara izquierda quede paralelo al piso.

## 7. RESULTADOS

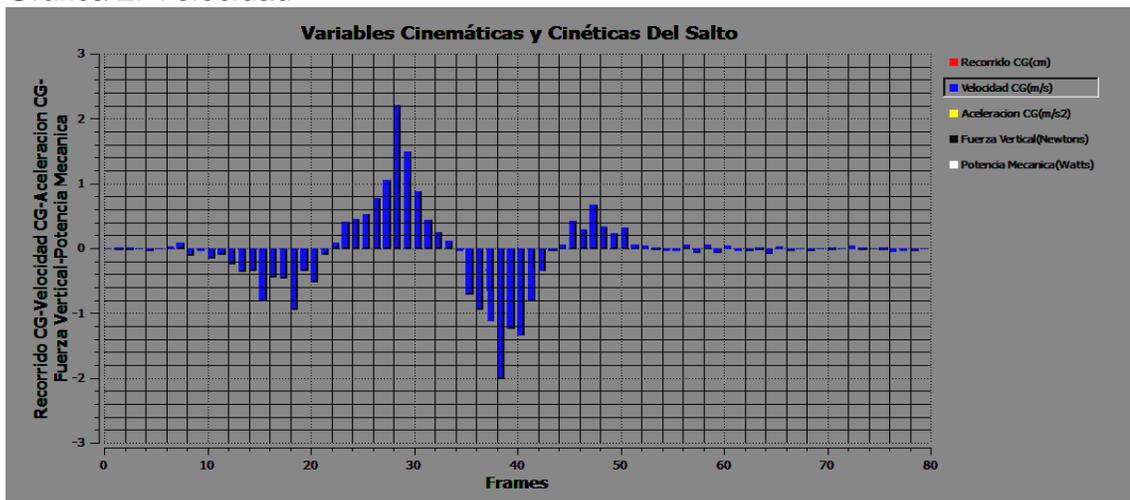
A continuación se muestran los resultados obtenidos por un deportista que pesó 80Kg y ejecutó un salto CMJ (salto con contra movimiento). Se puede observar las distintas gráficas de las variables cinemáticas y cinéticas del salto. Ver Gráficas 1, 2, 3, 4, 5.

Gráfica 1. Recorrido



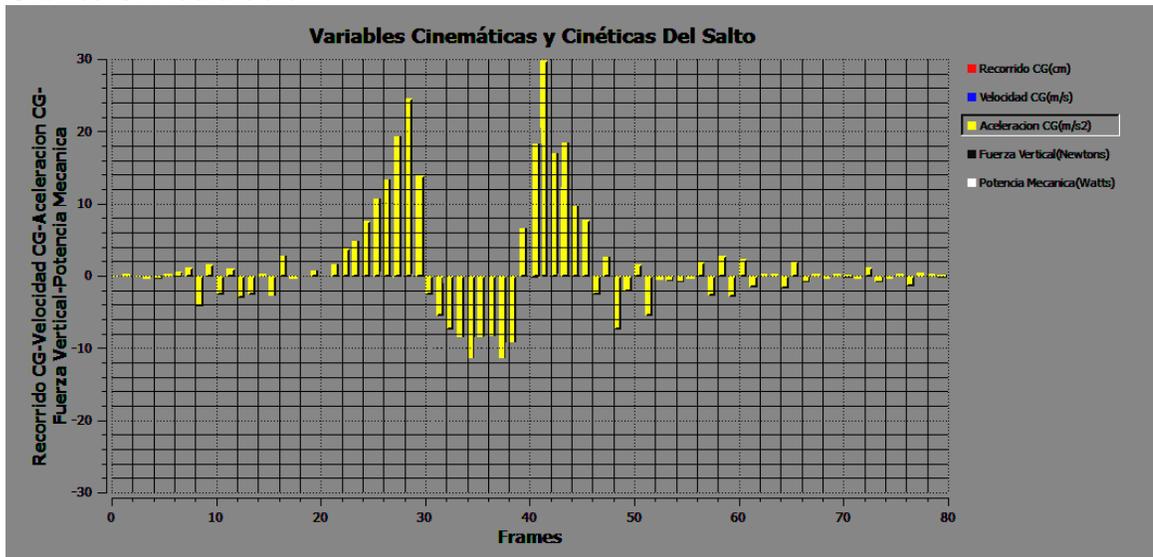
Se visualiza el trayecto realizado por el marcador que esta adherido a la cadera del deportista. Mostrando en el frames 22 el punto mínimo o altura mínima (punto más bajo de flexión para el impulso), luego una secuencia de subida hasta lograr la mayor altura (frames 34), después el ascenso es seguido del descenso con un punto de amortiguamiento mínimo encontrado en el frames 43 y por último se llega al punto de estabilidad o punto inicial.

Gráfica 2. Velocidad



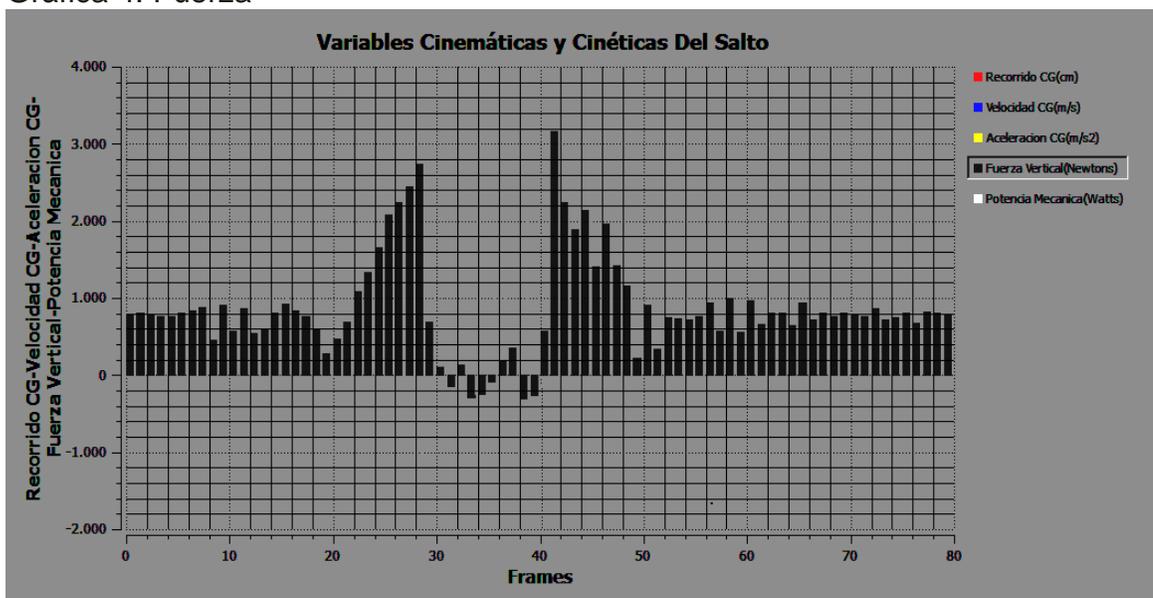
Lo más importante de la gráfica son los puntos que representan la velocidad máxima de subida y la velocidad máxima de caída, idealmente estos dos puntos deberían tener la misma magnitud y signo contrario, porque uno de ellos es la velocidad de ascenso máxima y el otro es la velocidad de descenso máxima, esto se puede ver en el frames 29 y 39.

Gráfica 3. Aceleración



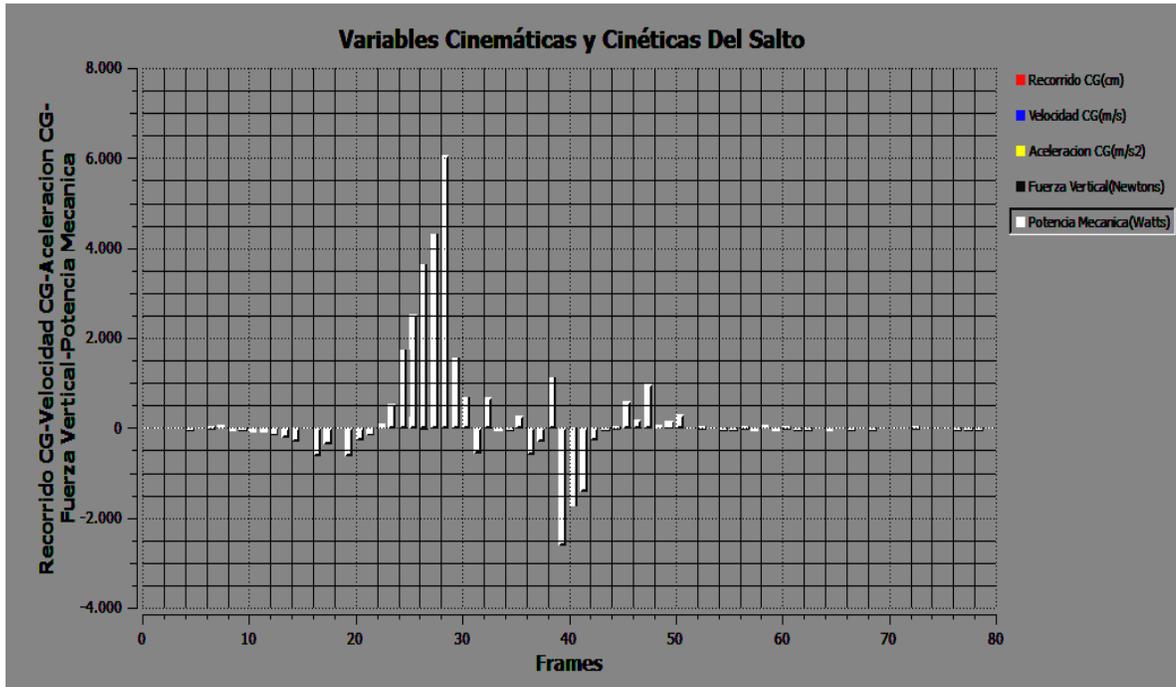
En el capítulo 8 se valida y se explica la gráfica de aceleración.

Gráfica 4. Fuerza



En el capítulo 8 se valida y se explica la grafica de fuerza.

Gráfica 5. Potencia



Siendo esta una grafica proporcional a la fuerza y velocidad hay dos puntos a tratar que son los mismos que en la grafica de velocidad (frames 29 y 39), el primero representa la potencia activa necesaria para impulsar al deportista hasta la mayor altura posible y la segunda muestra una potencia de reacción al momento de entrar en contacto con el piso en el descenso.

## 8. VALIDACIÓN DEL SISTEMA DE CAPTURA DE MOVIMIENTOS DIGITAL 3D

La razón principal de este capítulo es verificar que los datos que se obtienen del sistema de captura de movimientos satisfacen las necesidades planteadas y que sean equivalentes en comparación a otros sistemas comerciales o reconocidos. Como ejemplo de comparación se utilizaron los datos recogidos por el sistema de salto ChronoJump, los saltos fueron realizados en el laboratorio ALTIUS de la Universidad Surcolombiana perteneciente a la Facultad de Educación Física. El procedimiento usado para la validación fue poner a saltar 10 veces a una persona (con bastantes cualidades atléticas) con una diferencia de tiempo mínima de 3 minutos, ejecutando dos tipos diferentes de salto: el Squat Jump (SJ) y el Counter Movement Jump (CMJ).

Cuadro 1. Salto SJ (Squat Jump). Persona con un peso de 80 kg y altura 1,83 metros.

SJ(Squat Jump)					
Altura(cm)			Velocidad(m/s)		
ChronoJump	MoCap	Error %	ChronoJump	MoCap	Error %
33,820	32,32	4,641	2,576	2,61	1,316
40,160	38,78	3,559	2,800	2,7	3,704
24,742	23,95	3,307	2,203	2,14	2,934
28,492	27,52	3,532	2,365	2,28	3,715
32,022	31,36	2,111	2,500	2,44	2,459
Error Promedio (%)		3,430	Error Promedio (%)		2,826

En el Cuadro 1 se puede observar que en el salto SJ se obtienen dos variables cinemáticas del sistema Chrono Jump que son la altura alcanzada y la velocidad máxima, en comparación con el sistema MoCap realizado en este proyecto el porcentaje de error es pequeño y aceptable.

Cuadro 2. Salto CMJ (Counter Movement Jump). Persona con un peso de 80 kg y altura 1,83 metros.

CMJ (Counter Movement Jump)								
Altura(cm)			Velocidad(m/s)			Potencia de Lewis Peak 1974 (W)		
ChronoJump	MoCap	Error %	ChronoJump	MoCap	Error %	ChronoJump	MoCap	Error %
29,289	27,56	6,274	2,397	2,22	7,973	939,225	911,074	3,090
39,004	36,1	8,044	2,766	2,6	6,385	1083,854	1042,720	3,945
51,249	51,14	0,213	3,171	2,82	12,447	1242,389	1241,065	0,107
33,413	32,79	1,900	2,560	2,5	2,400	1003,166	993,768	0,946
42,690	40,93	4,300	2,895	2,73	6,027	1133,207	1110,286	2,064
Error Promedio (%)		4,146	Error Promedio (%)		7,046	Error Promedio (%)		2,030

Ahora en el Cuadro 2 se puede observar la inclusión de una variable cinética dada por el sistema de ChronoJump que es la Potencia de Lewis (diferente a la potencia mecánica), esta potencia solo es reportada cuando el salto es del tipo CMJ que puede ser calculada con la siguiente formula:

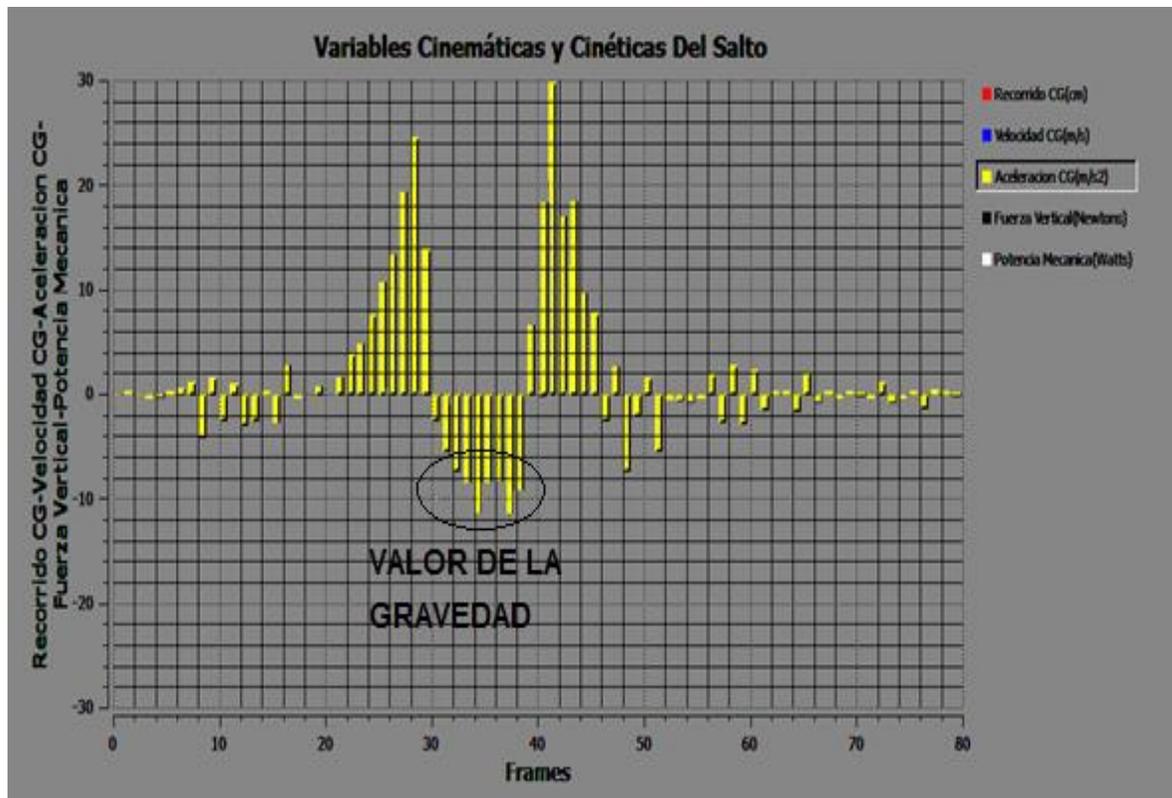
$$P = \sqrt{4,9} \times 9,8 \times (\text{PESO DEL CUERPO}) \times \sqrt{\text{ALTURA DEL SALTO EN METROS}}$$

Para uso de análisis y comparación se hace uso de la potencia de Lewis, según los datos tiene un porcentaje de error mínimo y suficiente para otorgar confiabilidad. El CMJ es un movimiento un poco más explosivo que SJ por esa razón se ve reflejado en el aumento de la altura y velocidad que a su vez refleja un incremento en el porcentaje de error de sistema MoCap con respecto al ChonoJump, para el caso de altura es un error aceptable, pero si se observa el porcentaje de error de las velocidades hay un incremento que puede verse mal a la hora de datar el sistema como fiable. Pero se debe recordar que el sistema de ChronoJump calcula la velocidad de despegue dependiendo de la altura alcanzada de una menara lineal y esto puede ser erróneo en algunos casos porque en la práctica se puede alcanzar la misma altura máxima con diferentes velocidades de despegue, pauta que depende de las características musculoesqueléticas del sujeto que efectúa el salto.

Por causa de no contar con una plataforma de fuerza en el laboratorio de ALTIUS, lugar donde se realizó las pruebas de salto, se acude a validar la variable de Aceleración de forma empírica y la Fuerza Vertical de forma gráfica en comparación con los datos obtenidos de la plataforma de fuerzas portable Quatro Jump de Kistler en la Universidad de Castilla de la Mancha y con la presencia de Dr. Xavier Aguado Jodár.

La variable cinemática de Aceleración se valida de forma empírica. En el momento que el sujeto comienza a caer después de haber alcanzado la altura máxima, la aceleración empieza a tomar el valor de constante de la gravedad y se mantendrá hasta que el deportista entre en contacto con el piso. Por la razón anterior se ve en la Gráfica 6 tomar el valor aproximado a la constante de la gravedad. Parte descrita en una elipse.

Gráfica 6. Validación empírica de la aceleración



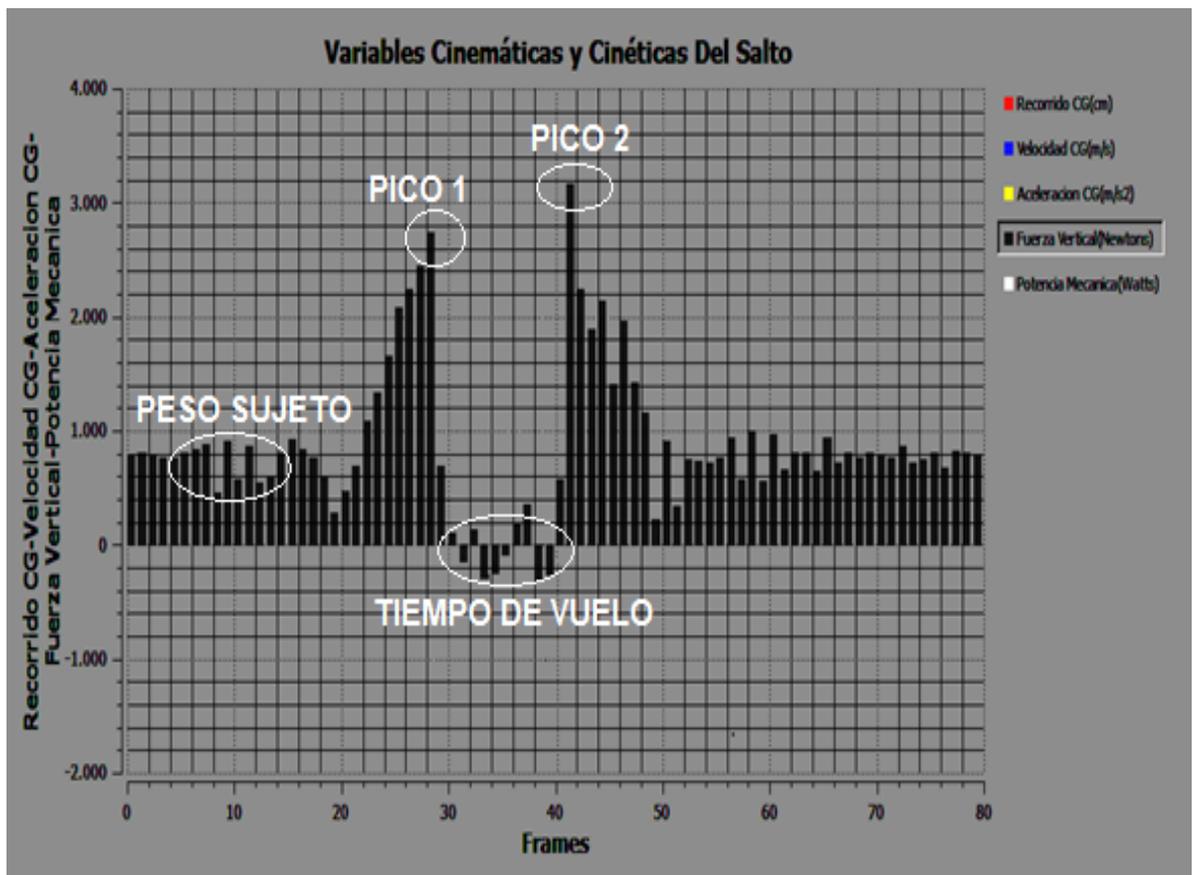
La aceleración por ser la segunda derivada de la altura muestra distorsiones al momento de encontrar los valores precisos del movimiento del salto, lo que crea una falencia a superar, en las conclusiones se hablara de cómo superar dicho percance.

La variable cinética de Fuerza vertical se valida comparando entre las gráficas de fuerza obtenidas por el sistema MoCap y la plataforma de fuerzas portable Quatro Jump de Kistler.

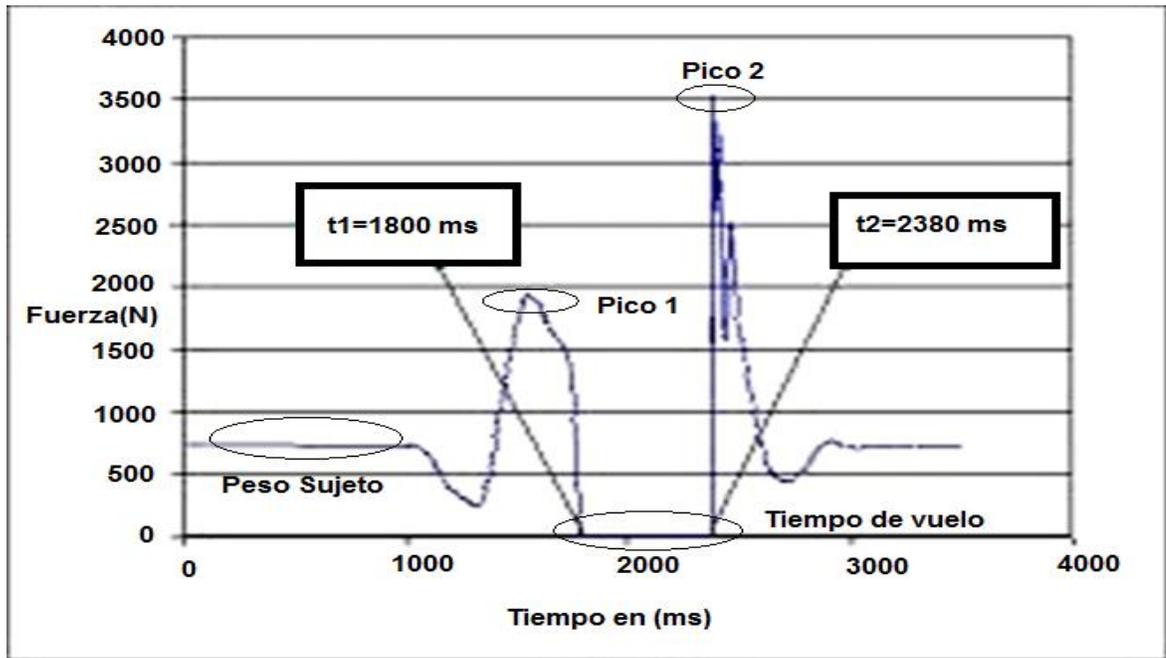
En cada una de las gráficas se describe cuatro elipses que encierran características importantes en el cumplimiento del salto. Ver Gráficas 7 y 8.

- El peso Sujeto: es la parte donde el sujeto se encuentra en reposo, siendo la fuerza resultante la que ejerce la gravedad sobre su cuerpo.
- El Pico 1: es el punto donde se aplica la mayor fuerza de tipo activa, cuando apenas va a despegar.
- El tiempo de vuelo: es el tiempo en el cual no se ejerce ninguna fuerza por que el sujeto se encuentra suspendido en el aire.
- El Pico 2: es el punto donde se aplica la fuerza máxima de tipo reactiva, debido a que entra en contacto con el piso luego del tiempo de vuelo.

Gráfica 7. Fuerza vertical del sistema MoCap



Gráfica 8. Fuerza vertical plataforma Quatro Jump de Kistler



Fuente:<http://www.rendimientodeportivo.com/N007/Artic032.htm>

Después de visualizar, analizar y comparar las gráficas de fuerza vertical se puede asumir que los datos del sistema MoCap desarrollado en este proyecto son semejantes a la forma de la gráfica de la plataforma de Quatro Jump de Kistler, aunque presenta un poco de error, que en las conclusiones se determinará como reparar dichas distorsiones. El método usado para validar estas dos últimas variables, fue tomado como guía del Trabajo de grado DISEÑO E IMPLEMENTACION DE UN SISTEMA DE VIDEOGRAMETRIA porque da muy buenos resultados.

## 9. CONCLUSIONES

Después de realizar determinado número de pruebas, conseguir los resultados y analizar conjuntamente todos los datos del proyecto se extraen las siguientes conclusiones:

- La realización del sistema MoCap 3D trae consigo ventajas muy notables, entre ellas se puede apreciar la transposición del mundo real a un entorno virtual sin perder la información tridimensional de los puntos a encontrar, todo en relación con la distancia verdadera con respecto a la cámara izquierda que representa el punto central o coordenada cero real, otra gran ventaja aplica en poder ver el movimiento del deportista desde cualquier ángulo y escala; después de ejecutar la calibración de las cámaras no requiere volver a calibrar las mismas al menos que se muevan una con relación a la otra, esto da como ventaja capturar varios movimientos sin estar recalibrando y por último se puede ejemplificar en una forma de diversión como lo es utilizar la información 3D adquirida para exportarla a programas de animación 3D para realizar cortometrajes, publicidad y/o videojuegos.
- El sistema ejecutado en tiempo real trae beneficios. Sin embargo, presenta inconvenientes al momento de adquisición, lo cual se traduce en una velocidad más baja de captura y procesamiento de las imágenes. En consecuencia, es el tiempo de ejecución el que debe trabajar el programa para procesar toda la información en el mismo instante. Considerando que el salto es un movimiento corporal de gran velocidad y aceleración, se concluye que el sistema presenta falencias a la hora de capturar el salto. A pesar de ello, se considera eficazmente válido para el tratamiento de videos con el mismo hardware utilizado en este proyecto o con la implementación de cámaras de alta velocidad (iguales o mayores a los 60 cuadros por segundo).
- El proyecto se diseñó e implementó, siguiendo los pasos adecuados para la captura y recolección de información del salto. Se incluyó un buen escenario (fondo y suelo negro), al igual que marcadores fluorescentes; también se desarrolló un software con una interfaz de usuario amigable, se utilizó un bajo presupuesto y se obtuvo las variables cinemáticas y cinéticas en un solo sistema; lo cual no acontece con otros sistemas que requieren de varias plataformas para conseguir las mismas variables obtenidas en el sistema de captura de movimientos de este proyecto.
- El sistema de captura de movimientos implementado, se diferencia de las plataformas de vuelo, en que se podrá visualizar la altura CG (centro de gravedad) del deportista con un cambio de tiempo que depende de la velocidad

de captura de las cámaras (FPS). Las plataformas de vuelo proporcionan solo la altura máxima alcanzada dependiendo del tiempo de vuelo.

- Aunque se reporte algún tipo de oclusión de los marcadores, el programa es adecuado para medir el recorrido CG (un máximo de 2 o 3 **frames** si se está en un movimiento rápido; unos cuantos **frames** de más, si es un movimiento lento y muchos si se encuentra estático). Este sigue bastante bien dicho recorrido, registrando un mínimo de error en la velocidad. No parece ser eficaz para calcular con precisión y sin ruido los datos a graficar de la aceleración, fuerza y potencia. Al parecer, es consecuencia de la jerarquía de las variables al ser datos derivados del recorrido CG. El ruido en la aceleración es más fuerte que el de la velocidad por poseer un nivel más alto de derivada. Como resultado, aumenta la fuerza y la potencia por tratarse de datos derivados de la velocidad y la aceleración.
- Este proyecto se realizó con la intención de mejorar el trabajo de grado: DISEÑO E IMPLEMENTACION DE UN SISTEMA DE VIDEOGRAMETRIA, mejoras que se pueden observar en la forma tridimensional de adquirir la información puntual de los marcadores, en la implementación de algoritmos de predicción, en la adquisición y procesamiento de las imágenes de manera sincronizada (uso en modo paralelo de un par de cámaras), rápida y de mayor calidad de imagen en tiempo real, visualización 3D tanto en tiempo real como en tiempo control y almacenamiento de la información en archivos .XML para su uso posterior en otras aplicaciones (animación 3D o muestra de resultados estadísticos). Con lo anterior se logró un mejor desempeño, fiabilidad, interactiva y utilidad del programa.

## 10. TRABAJO A FUTURO

Teniendo en cuenta las conclusiones anteriormente expuestas, los resultados obtenidos y algunas ideas surgidas al final del proyecto, se procede a listar ciertas tareas que podrían ser consideradas para trabajos posteriores:

- Utilizar cámaras de mayor velocidad de adquisición de cuadros por segundo. Si se cuenta con un mayor presupuesto, comprar cámaras integradas de visión estéreo prefabricadas, de mayor precisión y con velocidad de adquisición de cuadros por segundo, las cuales no presentan errores por el movimiento de una cámara con respecto a la otra. Ver Figura 37.

Figura 37. Cámaras visión estéreo



Fuente:<http://www.noticiasdot.com/publicaciones/gadgetmania/2011/08/21/sony-presenta-los-prismticos-videocmara-3d-dev-5-y-dev-3/>

- Reajustar la programación del seguimiento estéreo para operar con las nuevas herramientas gráficas CUDA, que se provee de unos núcleos gráficos que trabajan en paralelo con el procesador central de la computadora logrando mayor eficiencia al momento de ejecutar algoritmos complejos y robustos donde se vincule los gráficos, las cuales pueden ser enlazadas a la librería de OpenCV. Tiene un rendimiento en la velocidad de procesamiento en un mínimo de 10 veces mayor.

Si ninguno de los casos anteriores se presenta, es recomendable la grabación en video para procesarlos posteriormente. De esta manera se obtendrá una mayor velocidad de cuadros por segundo.

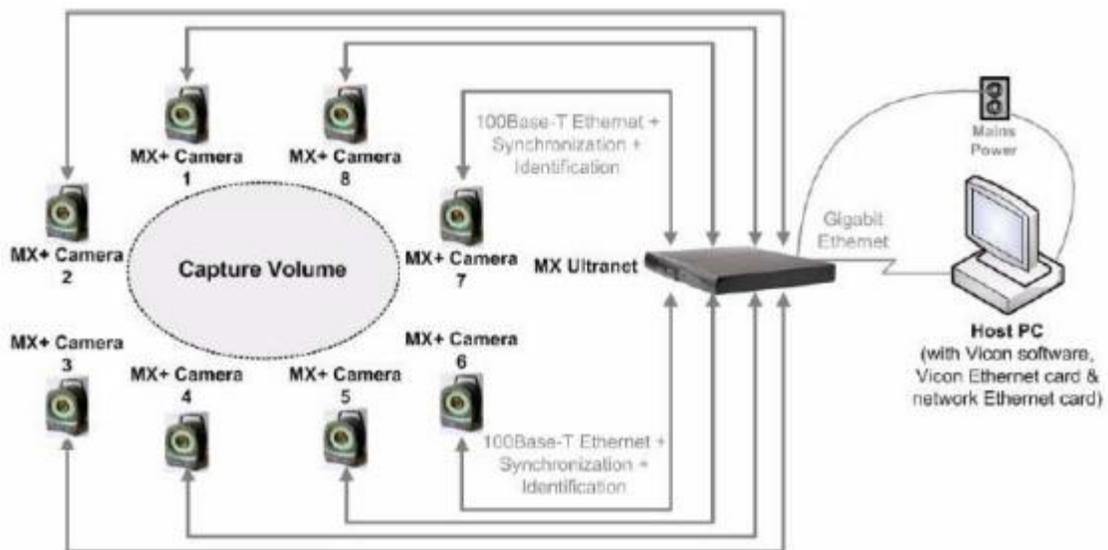
## MÚLTIPLES CÁMARAS:

En la actualidad existen diferentes sistemas de captura y análisis de movimiento que tienen aplicaciones en campos médicos, deportivos y cinematográficos. Uno de los sistemas que más sobresale en el mercado es el VICON MX.

El sistema Vicon MX es el sistema óptico más avanzado de captura de movimientos disponibles. Los principales componentes del sistema Vicon son las cámaras, el módulo de hardware de control, el software (para analizar y presentar los datos) y el equipo para ejecutar el software. MX ofrece una mayor precisión, mayor rendimiento y mayor funcionalidad que sus versiones precedentes.

Cada sistema Vicon MX incluye al menos un MX Giganet para proporcionar energía y comunicación de datos hasta con 10 cámaras y otros dispositivos. El MX Giganet también gestiona el flujo de datos a la computadora que ejecuta el software que se utilizará para analizar los datos. Si el sistema tiene más de diez cámaras necesitará otra Giganet por cada diez cámaras adicionales.

Figura 38. Arquitectura básica vicon mx



Fuente: <http://bibing.us.es/proyectos/abreproy/11944/fichero/Cap%C3%ADtulo+3.pdf>

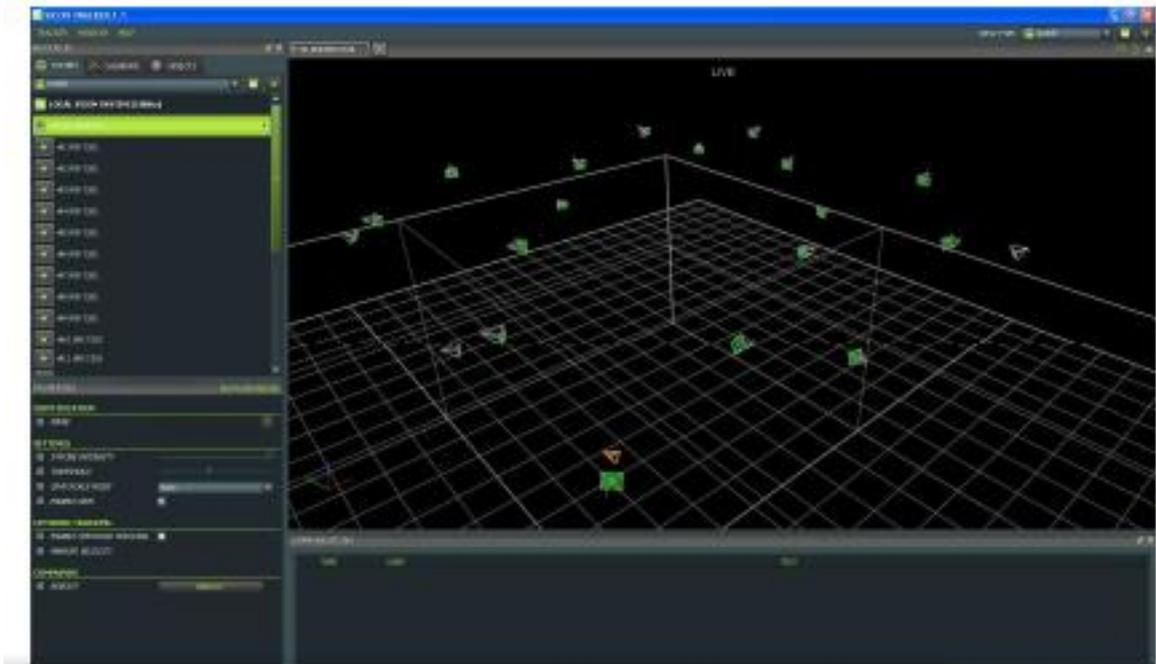
El software con el cual se analizan los datos captados por las cámaras está albergado en un PC, que se conecta al MX Giganet a través de un puerto gigabyte. Según la necesidad de la aplicación se usará un software u otro. En este caso se usarán los programas Tracker y Nexus para la necesidad de seguimiento en vuelo de quadrotors.

ViconTracker es una potente solución de seguimiento de objetos que proporcionan una exactitud sin precedentes para la integración de datos en aplicaciones 3D.

Tracker ha sido diseñado para aplicaciones tan diversas como: el seguimiento de robots, ingeniería de factores humanos, la optimización del método de diseño, ingeniería virtual y pre visualización.

En la figura se presenta la interfaz de Tracker donde vemos que se divide en dos partes. La parte de la derecha es la dedicada a la configuración de las cámaras, al calibrado y manejo de objetos. En la parte de la izquierda se representa la imagen que capturan las cámaras. Ver Figura 39.

Figura 39. Interfaz de tracker



Fuente: <http://bibing.us.es/proyectos/abreproy/11944/fichero/Cap%C3%ADtulo+3.pdf>

## BIBLIOGRAFIA

ACEVEDO SUAREZ, DerlyYohana; HINCAPIE MUÑOZ, Francia Milena; SANCHEZ PIZARRO, Jorge Leandro. “valoración de la manifestación reactiva de la fuerza de los miembros inferiores a las integrantes de la selección Antioquia de voleibol categoría junior rama femenina”. (2008)

Blanes F.; Jiménez L.M.; Puerto R.; Ñeco R.P.; Reinoso O.” reconstrucción tridimensional de escenas con un par estereoscópico de cámaras”. (2008).

Bosco, C. “La valoración de la fuerza con el test de Bosco” Editorial Paidotribo. Barcelona.1994.

Gary Bradski and Adrian Kaehler. OpenCV: Computer vision with the OpenCV library. O'Reilly.(2008)

GUZMAN CENTANARO, Douglas Humberto.”Integración de OpenGL en Qt”.(2011)

Intel Corporation, Open Source Computer Vision Library Reference Manual. (2001)

IZER GONZALES, Joseph. Estudio experimental de métodos de calibración y autocalibración de cámaras.(2003)

PERDOMO ORTIZ, Diego Fernando.” Diseño e implementación de un sistema de videogrametría digital para el análisis de saltabilidad en deportistas”.(2010)

TARLEA JIMENEZ, Jorge. “Sistema de posicionamiento de objetos mediante visión estéreo embarcable en vehículos inteligentes”. (2009)

Zhengyou Zhang. A flexible new technique for camera calibration. (1998)

<http://bibing.us.es/proyectos/abreproy/11944/fichero/Cap%C3%ADtulo+3.pdf>

[http://chronojump.org/tests\\_es.html](http://chronojump.org/tests_es.html)

<http://sabia.tic.udc.es/gc/Contenidos%20adicionales/trabajos/Peliculas/Mocap/tecnol.htm>

[http://www.uclm.es/profesorado/xaguado/INVESTIGACI%D3N/Curriculum%20laboratorio/articulos%20pdf/LaraAJ\\_2005%20AMD%20Pot%20f%F3rmulas.PDF](http://www.uclm.es/profesorado/xaguado/INVESTIGACI%D3N/Curriculum%20laboratorio/articulos%20pdf/LaraAJ_2005%20AMD%20Pot%20f%F3rmulas.PDF)

## ANEXO

### ANEXO A. SISTEMA CHRONOJUMP

Uno de los métodos más eficientes en la actividad deportiva en cuanto al salto se refiere, es el sistema chronojump. Consiste en hallar una serie de parámetros importantes a la hora de realizar un salto. Entre los más destacados cabe mencionar los siguientes:

- Saltos simples: Salto libre, SJ, SJ (con peso extra), CMJ, CMJ (con peso extra), ABK, DJ (con uso de brazos), DJ (sin uso de brazos), Rocket, Batida, Batida (con peso extra).
- Multisaltos: RJ (limitado por saltos), RJ (limitado por tiempo), Ilimitado, Hexágono, Triple salto. Los multisaltos tienen feedback para ayudar a controlar el test.

Los parámetros que mide el sistema son:

Altura de salto, tiempo de vuelo, tiempo de contacto (en algunos tests) y la potencia. Estadísticas y gráficas que comparan los distintos tests.

Figura 40. Test de salto

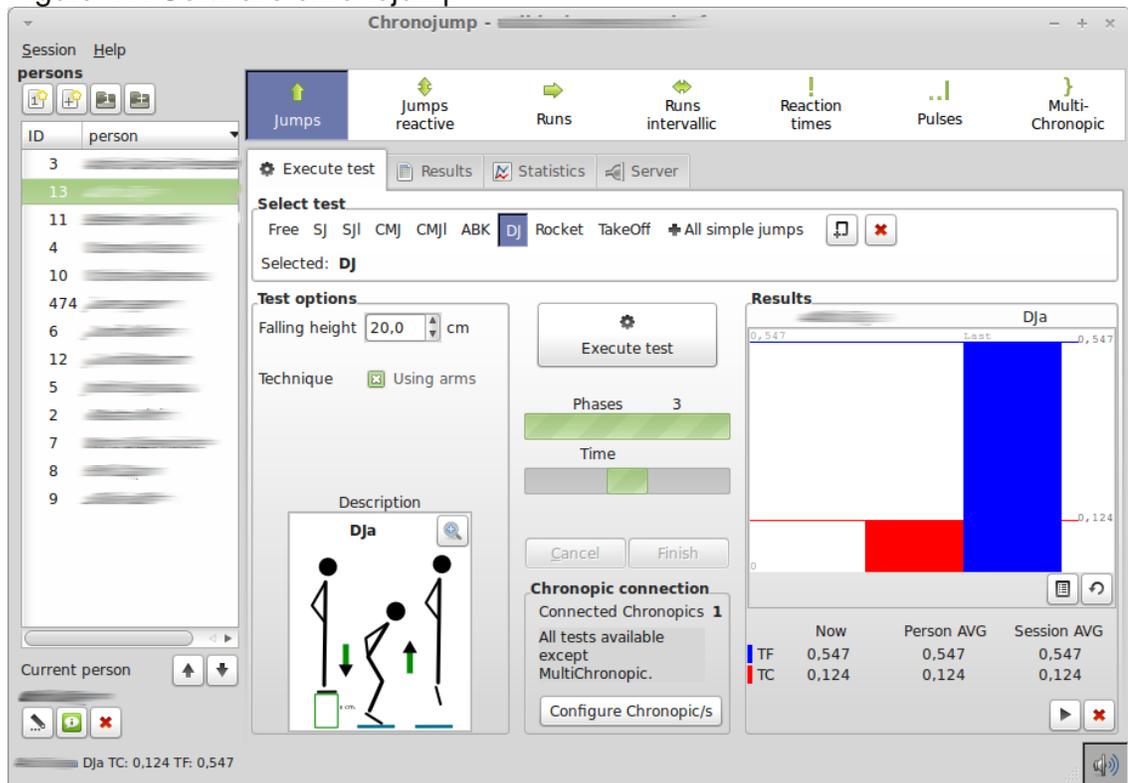


Fuente: <http://personaltrainers-es.blogspot.com/2011/05/5-lecciones-de-entrenamiento-de-yuri.html>

Los productos que contiene el sistema chronojump son:

1. Software Chronojump

Figura 41. Software chronojump



Fuente: [www.chronopicjump.org](http://www.chronopicjump.org)

El software Chronojump funciona en Windows y Linux, y está disponible en 7 idiomas.

El software lee los datos del microcontrolador Chronopic 3 y registra tests temporales como saltos, carreras, ritmos y tiempos de reacción. Los tests, sujetos y sesiones se guardan en una base de datos. El software incluye la batería de tests de Bosco y otros tests. Permite que los usuarios diseñen sus propios tests.

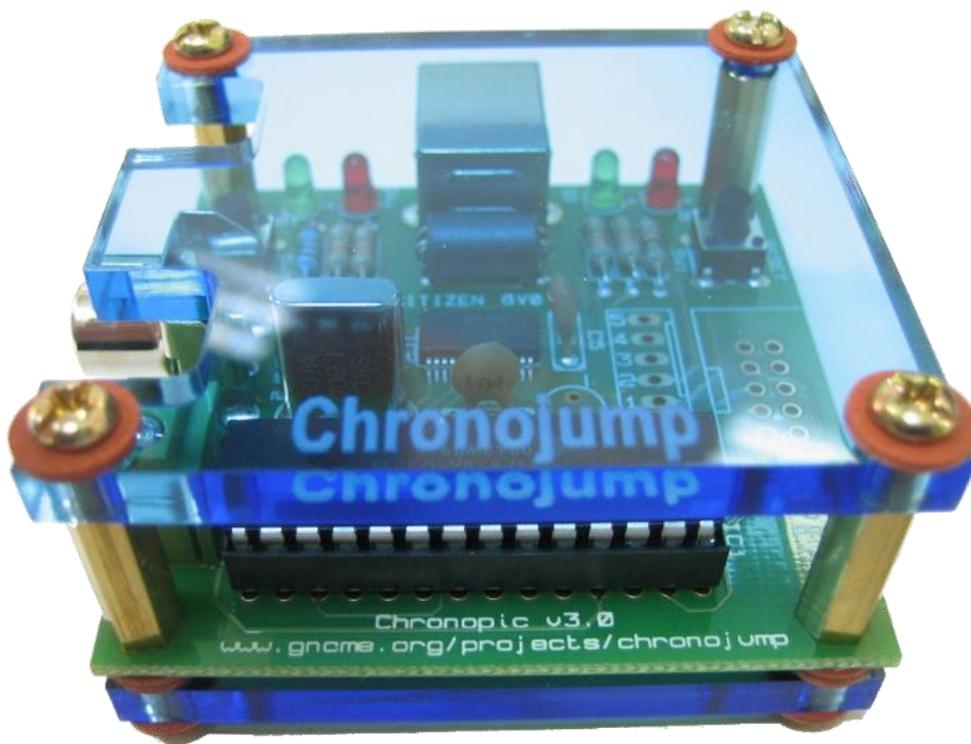
Los tests repetitivos como los multisaltos, incorporan feedback visual y auditivo con la finalidad de ayudar en el control del entrenamiento. Se pueden registrar todos los tests con una webcam. Se incluyen estadísticas y gráficas que permiten la generación de informes y exportación de los datos. Finalmente, si se desea, se puede conectar a un servidor para compartir los datos con otros usuarios respetando la privacidad de los ejecutantes.

Chronojump ganó el premio al 'mejor software libre' en el 2007, en la categoría de

Educación en la competición internacional “Les Trophées du Libre” por la fundación CETRIL.

## 2. Chronopic 3

Figura 42. Chronopic 3



Fuente: [www.chronopicjump.org](http://www.chronopicjump.org)

Chronopic 3 lee los tiempos detectados en plataformas de contactos, fotocélulas o pulsadores y envía los datos al ordenador. Puede elegir Chronopic 3 en dos versiones de firmware en función del mínimo tiempo detectable:

50ms: ideal para saltos en plataformas (ayuda a prevenir falsos saltos en caso de errores en la técnica)

10ms: ideal para carreras entre células fotoeléctricas

Cuando lo desee puede cambiar de firmware con el software Chronopic-firmware-record que se instala con Chronopic.

En ambos casos la precisión es cercana a 1ms. El valor 50 o 10 se refiere al tiempo mínimo necesario para que algo sea considerado como salto o carrera. Esta precaución (50ms para saltos, 10 ms para carreras) permite solucionar problemas de señales eléctricas poco limpias en los dispositivos de registro (como en una plataforma de contactos elaborado por el usuario y que disponga de

múltiples conexiones eléctricas). En definitiva, para que un salto sea detectado se requerirá que sea como mínimo de 50ms, pero el sistema diferenciará sin problemas entre un salto de 50ms y uno de 51ms.

### 3. Plataforma de contactos

Figura 43. Plataforma



Fuente: [www.chronopicjump.org](http://www.chronopicjump.org)

Plataforma de alta calidad resistente a la flexión (fibra de vidrio) y con buena conductividad.

A diferencia de las plataformas de varillas, la plataforma rígida produce contacto en toda su área evitando falsos contactos o caídas en zonas muertas.

Medidas: Din-A2, Din-A3, Din-A4.

## ANEXO B. GUIA DE USUARIO DEL SOFTWARE

Figura 44. Interfaz



El programa se desarrolló de forma secuencial en coincidencia con lo explicado en todo libro de tesis, con las siguientes funciones:

- **BOTON DE CALIBRACION INDIVIDUAL:** En esta fase del programa se realiza un tipo de calibración de cámara para encontrar los parámetros intrínsecos y de distorsión de la misma; cuenta con lo siguiente:
  1. Botón Play: Inicia la extracción de imágenes de la cámara indicada.
  2. Botón Pause: Pausa la extracción de imágenes.
  3. Botón Stop/Reinicio: Reinicia la calibración individual, pudiendo cambiar la cámara de adquisición;
  4. Botón Limpiar Texto: Quita por completo todo texto que haya en el visor de texto.
  5. Índice de la cámara: Representa el número de cada cámara para su debida obtención.
  6. Visor de texto: Muestra texto de advertencia o del estado de la calibración.
  7. Visor de cámara: Muestra el video obtenido de la cámara.
  8. Barra de progreso: Muestra el progreso en la captura de los puntos del damero.

- **BOTON DE CALIBRACION ESTEREO:** En esta fase se procede a calibrar el par de cámaras de manera estéreo, encontrando una matriz  $Q$  que relación la cámara derecha con respecto a la izquierda en cuestión de translación y rotación, al igual que los parámetros intrínsecos, y por último unas matrices de corrección de imagen para simular un montaje de cámaras en forma paralela ideal. Todo esto para después triangular y obtener las posiciones tridimensionales de los objetos (marcadores). Los botones usados son los mismo que en calibración individual con la diferencia que hay dos visores de cámara.
- **BOTON SEGUIMIENTO ESTEREO:** Esta parte realiza la grabación del salto y visualización 3D en tiempo real del deportista. Los botones usados son los mismos que en la etapa de calibración estéreo más los siguientes botones:
  1. Botón Grabar: Graba la secuencia de salto de forma textual, guardando las posiciones tridimensionales de cada frames de los 80 en total, estas posiciones representan a los marcadores incluidos al deportista.
  2. Botón 3D: Muestra el movimiento del deportista en forma animada 3D en tiempo real.
  3. Sliders de Rango: Sliders para cambiar el valor de segmentación, para adquirir solamente los marcadores y así evitar interferencias.
- **BOTON TIEMPO CONTROL:** Es una representación tridimensional real del salto grabado en la fase de seguimiento estéreo con control en la reproducción (hacia adelante o atrás) , al igual que la visualización de unos parámetros del salto:
  1. La altura de la cadera, rodilla y tobillo en cada frames con respecto a la altura inicial de cada uno.
  2. Posición tridimensional en cada frames de todos los marcadores con respecto a la cámara izquierda.
  3. Angulo de la cadera con respecto al tobillo en todos los frames.
  4. Distancia tridimensional entre los marcadores.
  5. Tiempo real en cada frames según lo grabado.
  6. Tiempo de vuelo del salto.
  7. Altura máxima y mínima alcanzada por el deportista.
  8. Frames promedio alcanzado durante la grabación.
  9. El número de cada frames.

**BOTON ESTADISTICA:** En esta última fase se muestran las gráficas de recorrido, velocidad, aceleración, fuerza y potencia por medio de barras, capaces de solaparse entre sí, pero manteniendo la escala de la más grande en valor. En todas las fases al igual que en la parte principal se cuenta con menú flotante de ayuda rápida y una acción de cerrar.