

Implementación de un sistema de procesamiento para análisis de video en biomecánica.

A Guchin, G Pereira, G Ottado, M Ramos y J Cardelino.

Instituto de Ingeniería Eléctrica, Facultad de Ingeniería, Universidad de la República,
Uruguay.

E-mail: proyecto.iie.biomecanica@gmail.com

Abstract. This paper proposes a processing system applied to optical motion capture based on markers that facilitates human biomechanical analysis. The system consists of separate blocks, giving an overall solution which allows to easily modify or replace components. Guidelines are described for generating a virtual laboratory to obtain a database with sequences of synthetic videos and explore the characteristics required for a laboratory based on optical motion capture system. A set of metrics is presented to measure the performance of each block and the overall system. Tests on the implemented software reflected that it has an average accuracy of less than one centimeter on sequences obtained in controlled environments. These results are acceptable considering that the algorithms used in each block are of low complexity and can be optimized.

1. Introducción

En distintos ámbitos académicos y profesionales se necesita realizar un estudio del movimiento del cuerpo humano. Esta tarea implica registrar el movimiento de miembros o articulaciones en el espacio y su correspondiente evolución en el tiempo. Algunos ejemplos de distintas áreas que ilustran estas necesidades son a *nivel asistencial* en el área de *fisioterapia*, *investigación académica en biomecánica*, *medidas de performance* en el deporte de alto nivel y *Animación 3D*, entre otras. Los ejemplos mencionados definen distintos casos de uso con características disímiles, de manera que la búsqueda de una solución única que abarque las necesidades particulares de todos ellos resulta compleja. Uno de los métodos ampliamente utilizados para realizar estos estudios son los sistemas ópticos de captura de movimiento. Dichos sistemas construyen un modelo digital del movimiento a partir de secuencias de video capturadas con cámaras.

El objetivo de este trabajo es el desarrollo de una aplicación de software básica y funcional de código abierto para un sistema óptico de captura de movimiento, que brinde solución a las necesidades anteriores abarcando el problema en forma general, pero que pueda extenderse a otros casos de uso. Contar con este tipo de herramientas es fundamental para las necesidades de los equipos de profesionales, cuya alternativa son productos comerciales de alto costo.

El software desarrollado pretende obtener las coordenadas espaciales de un número de puntos de interés sobre un persona a partir de imágenes de video capturadas bajo ciertas condiciones controladas. Se asume que dichas condiciones consisten en que el individuo porte un traje oscuro y marcadores blancos en un ambiente con iluminación adecuada, se filma con múltiples cámaras a lo largo del tiempo, se lleva esta información a la computadora y se procesa para obtener la posición 3D de los marcadores a lo largo de toda la secuencia.

Trabajando a partir del algoritmo propuesto por Herda [1], se propone una versión reproducible del mismo con medidas de performance sobre cada etapa, así como una metodología de diseño de un laboratorio virtual desarrollando en conjunto un sistema completo de prueba.

1.1. Estado del arte

Centrados en el desarrollo de un sistema con precisión aceptable pero que además fuera accesible, orientamos la búsqueda hacia sistemas que emplean cámaras convencionales y marcadores. Existen artículos que describen el estado del arte de diferentes técnicas de captura de movimiento pero no se ha encontrado mucha bibliografía que resumiera el estado del arte de sistemas que empleen cámaras convencionales en particular. Sí se ha encontrado en cambio, diversos artículos que describen implementaciones, totales o parciales, que utilizan esta metodología. Dentro de esa bibliografía se encontró la tesis elaborada por Herda [1]. La misma se utilizó como base para nuestro desarrollo dado que el sistema que propone posee las características buscadas y es ampliamente citado. No obstante, dicha tesis presenta ambigüedades en la descripción de la metodología utilizada en algunos procesos.

En cuanto a las soluciones disponibles actualmente, por un lado se tienen sistemas de captura de movimiento comerciales que destacan por su performance, como por ejemplo: Vicon [2], OptiTrack [3], PhaseSpace [4], Qualisys [5], MotionAnalysis [6]. La mayoría de estos sistemas utilizan cámaras infrarrojas, lo que permite una detección más eficaz de marcadores pero con la desventaja de su elevado costo. Por otro lado existen alternativas de software libre. Si bien no se han encontrado implementaciones que realicen íntegramente todas las etapas de un sistema de captura de movimiento, sí existen implementaciones que permiten realizar algunas de estas etapas. Por ejemplo, el software *Kimovea* [7] realiza el seguimiento de puntos en imágenes de video, esta herramienta permite un análisis fundamentalmente cualitativo del movimiento orientado al deporte, por lo que no es aplicable a nuestros objetivos. Existen además librerías como *OpenCV* y *toolboxes* escritos en *Matlab*, algunas de estas herramientas fueron utilizadas en el sistema desarrollado.

Se buscaron además imágenes de secuencias de captura de movimiento con las cuales poder desarrollar y probar nuestro sistema de procesamiento, encontrándose numerosas bases de datos. Dichas bases fueron diseñadas para capturas con cámaras infrarrojas por lo que las condiciones de laboratorio empleadas en las mismas no son las adecuadas para cámaras convencionales. Por lo tanto las imágenes de dichas bases de datos no fueron utilizadas como imágenes de prueba para nuestro sistema, sin embargo si se utilizó información de las trayectorias 3D del movimiento de los marcadores utilizados en dichas bases para generar un laboratorio de captura virtual el cual se describe en la sección 2.

2. Generación de secuencias

Con el fin de implementar, testear y comparar los distintos tipos de algoritmos desarrollados por el sistema, es deseable poseer múltiples secuencias de videos 2D de movimiento obtenidas con cámaras situadas en un entorno 3D cerrado, previamente acondicionado. Así como también contar con el correspondiente ground truth 2D y 3D¹ de los datos de movimiento disponibles y la información de calibración de las cámaras utilizadas para efectuar las capturas.

2.1. Características de Laboratorio

Para facilitar el posterior procesamiento es necesario cuidar la relación entre algunas variables a la hora de diseñar un laboratorio para un sistema de captura óptica basado en marcadores.

¹ Por ground truth se entiende a aquella información que se considera verdadera, en este caso referida a las trayectorias 2D y 3D del movimiento de interés. Por lo tanto las pruebas de performance de las distintas partes del sistema son contrastadas respecto al ground truth.

Cámaras. La resolución espacial de los datos en el procesamiento condiciona la resolución de la cámara, la resolución temporal para la marcha requiere como mínimo 30 cuadros por segundo, con tiempos de obturación de al menos $1/2000$ s, esto último permite evitar efectos de distorsión debidos a falta de nitidez.

Marcadores. El color de los marcadores debe contrastar claramente con la vestimenta y el fondo del espacio de captura, se recomienda una forma esférica. Si el fondo del espacio de captura es negro, opaco al igual que la vestimenta del sujeto a capturar, los marcadores pueden ser blancos, no pulidos de manera que su reflejo sea difuso, para no generar variación de tonos sobre los mismos. En capturas con cámaras ubicadas a menos de 12 metros del movimiento a relevar, un tamaño aceptable para el marcadores es de 3 cm de diámetro.

Vestimenta. Debe ser ajustada, para desprestigiar fluctuaciones en la posición de los marcadores y preferiblemente de igual color que el fondo.

Iluminación. debe ser uniforme, si se utiliza iluminación artificial con focos puntuales es habitual colocar pantallas difusoras delante de los focos.

Espacio de captura. debe contrastar con los marcadores, y sus dimensiones varían según el tipo de marcha a relevar. En caso de marcha rectilínea sobre una plataforma de $3\text{ m} \times 5\text{ m}$ se encuentra que 4 son el mínimo número de cámaras que permiten relevar el movimiento de manera satisfactoria. Mientras que en el caso de la marcha libre sobre una plataforma circular de 5 m de diámetro, se recomienda la utilización de al menos 8 cámaras. Por más detalles sobre la distribución de las cámaras consultar [8].

2.2. Laboratorio Virtual.

Utilizando la suite de animación 3D gratuita y de código abierto de *Blender*, se genera un laboratorio de captura de movimiento virtual, donde se obtienen secuencias de movimiento sintéticas con sus respectivos videos a partir de un modelo virtual 3D. El esqueleto del modelo contiene la información de movimiento, el mismo se obtuvo de la base de datos *MotionBuilder-friendly versión* ofrecidas por *cgspeed* [9], donde se cuenta con las fuentes BVH que provienen de capturas de movimiento real de *Carnegie Mellon University Motion Capture Database* (C.M.U.). Si bien las secuencias de video obtenidas son lo único necesario para el análisis posterior, al generar dichas secuencias a través de un entorno virtual controlado, permite probar múltiples configuraciones contando en cada caso, con la información exacta del ambiente de captura. El laboratorio virtual utilizado en este trabajo posee 17 cámaras dispuestas uniformemente sobre un ambiente rectangular de $10 \times 15\text{ m}$ permitiendo probar diferentes combinaciones de captura a la hora de generar secuencias. Para iluminar el laboratorio se utilizan 8 focos puntuales de luz omnidireccional sobre los límites del mismo, rodeando la escena y a un nivel de 3 metros de altura. Cuidando que ninguno de los focos sea tomado directamente por las cámaras y los marcadores se iluminen correctamente.

Blender permite extender su funcionalidad a través de programas en Python, automatizando varias etapas en el desarrollo de nuevas secuencias, así como la exportación de información a otros lenguajes. Por más detalles de la implementación del laboratorio virtual así como de la generación de secuencias consultar [8].

3. Implementación

Un sistema de captura de movimiento con las características necesarias para cumplir el objetivo de este proyecto debe implementar cuatro bloques generales: *calibración, detección de marcadores, reconstrucción y seguimiento*. En la Figura 1 se muestra un esquema del sistema a implementar, cada bloque verde indica la salida de una etapa siendo a su vez la entrada del bloque siguiente.

Es importante destacar la independencia entre bloques, permitiendo modificar u optimizar fácilmente el sistema en etapas futuras. A continuación se describe el funcionamiento de cada etapa del proceso de captura de movimiento, así como su implementación.

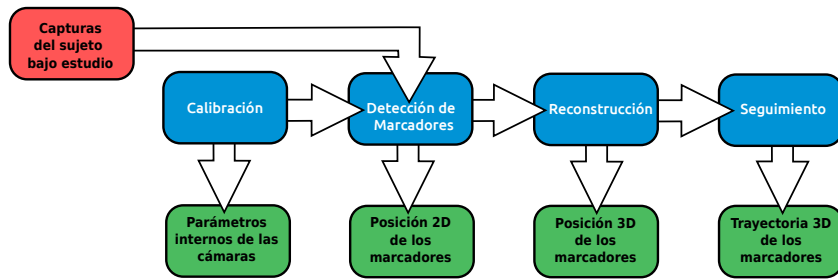


Figura 1: Diagrama de bloques del sistema completo.

3.1. Calibración

Para establecer una metodología de calibración válida para la configuración de cámaras con las que se diseñó el entorno virtual descrito en la sección 2, se probaron distintas implementaciones existentes, evaluando dos toolbox elaborados en *Matlab*. La metodología de calibración fue simulada en *Blender* mediante *scripts Python* y las imágenes resultantes procesadas con dichos toolbox. La descripción de la metodología y las simulaciones se detalla en [8].

Uno de los toolbox utilizados es el *Automatic Multi-Camera Calibration Toolbox (amcctoolbox)* [10], el cual utiliza como objeto de calibración un damero. Este método, con resultados de buena precisión [11], puede no ser suficientemente flexible para un sistema de muchas cámaras ya que, entre otras cosas, es necesaria la intervención manual en algunos casos.

El otro toolbox utilizado es el *Multi-Camera Self-Calibration Toolbox* [12]. Este método consiste en capturar el movimiento de una fuente puntual de luz que recorra el volumen de trabajo. Para cada cuadro se tiene un punto 3D en el espacio en una posición distinta y en cada una de las cámaras su correspondiente proyección. El error de re-proyección promedio obtenido es menor a 0.13 píxeles para todas las cámaras. Este método plantea una forma simple de calibrar un conjunto de muchas cámaras adecuado para el sistema de 17 cámaras del laboratorio virtual desarrollado en *Blender*.

3.2. Detección de marcadores

El bloque de detección de marcadores, se puede dividir en dos partes: la *segmentación* y el *filtrado de objetos*. El algoritmo realiza la detección siguiendo el siguiente proceso:

- (i) Obtención de cada cuadro de la entrada de video.
- (ii) Se toma un cuadro y se segmenta utilizando umbralización de Otsu.
- (iii) A partir de la imagen segmentada, se identifican los marcadores.
- (iv) Se escribe la posición de los marcadores detectados en un archivo XML.
- (v) Se toma el siguiente cuadro y se repite el proceso a partir del paso 2.

3.2.1. Descripción de las etapas de detección. El bloque de *segmentación* utilizó umbralización generando umbrales con el método de Otsu[13] de tres clases.

La etapa de *filtrado* no es más que una clasificación de los objetos segmentados. Dado que los objetos a detectar tienen formas relativamente sencillas (círculos blancos sobre fondo oscuro) y las condiciones de laboratorio son controladas al realizar la captura, esta etapa no requirió implementar algoritmos muy complejos. En particular, se implementó un detector de objetos circulares en base a momentos geométricos[14] y un filtro según el área de los mismos.

3.2.2. Resultados. En la etapa de segmentación se observó que el resultado obtenido depende fuertemente de las condiciones de captura y del umbral calculado. Si las condiciones de captura se apartan de las definidas en la sección 2.1, los resultados pueden no ser satisfactorios. Sin

embargo, si las capturas se hacen dentro de las condiciones establecidas, los resultados obtenidos son aceptables, como se observa en la Figura 2.

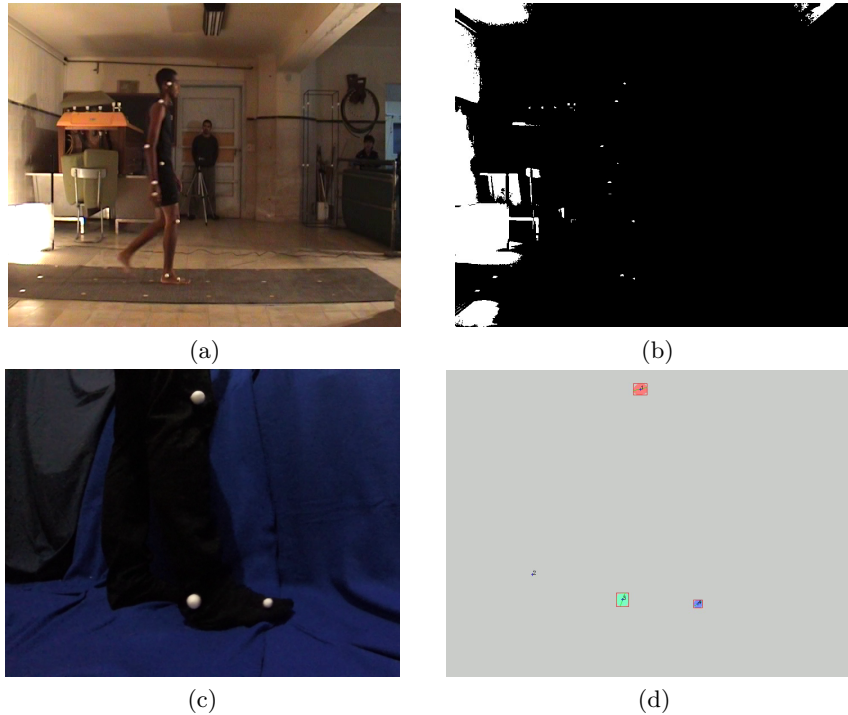


Figura 2: (a) Imagen original de una secuencia fuera de las hipótesis de captura. (b) Resultado de la segmentación fuera de las hipótesis de captura. (c) Captura original de una secuencia real bajo las hipótesis de captura. (d) Marcadores detectados.

3.3. Reconstrucción

A la salida del bloque de detección de marcadores se tiene, para cada cámara y para cada cuadro de una secuencia, un conjunto de coordenadas (x, y) que ubican la posición en la imagen de aquellos marcadores que fueron detectados. El proceso de reconstrucción consiste en obtener la posición de los marcadores en el espacio, a partir de la posición de los marcadores sobre las retinas de al menos dos cámaras. El proceso de reconstrucción que se presenta fue inspirado en el trabajo de Herda y consiste en tres pasos fundamentales:

- (i) Encontrar la correspondencia entre puntos en cámaras diferentes.
- (ii) Seleccionar la mejor correspondencia.
- (iii) Reconstruir y verificar en el resto de las cámaras.

3.3.1. Algoritmo El algoritmo implementado recibe como entrada los puntos 2D de los marcadores detectados y devuelve como salida los puntos 3D reconstruidos. En la Figura 3 se presenta un diagrama del algoritmo.

Asociar puntos 2D Este bloque recibe como entrada las coordenadas de los puntos detectados en cada una de las cámaras, parámetros de las mismas tales como sus matrices de proyección y devuelve para cada punto una lista ordenada por relevancia, de las asociaciones existentes con puntos en otras cámaras.

Mejor asociación. A partir de la lista con asociaciones entre pares de cámaras es necesario

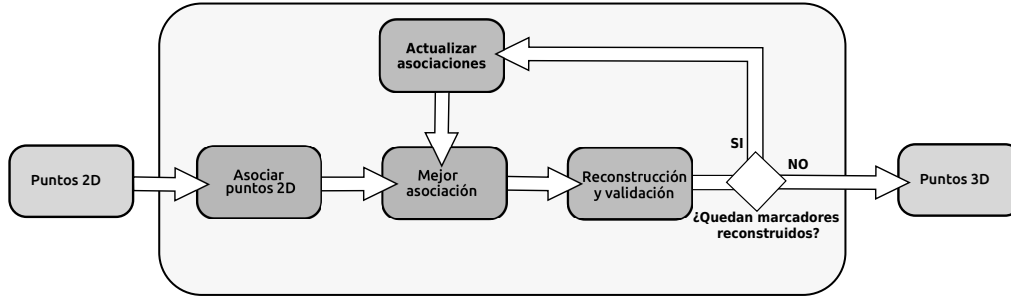


Figura 3: Diagrama de bloques del algoritmo de reconstrucción.

elegir aquella que posea mayor probabilidad de conformar la pareja de imágenes correspondiente a la proyección de un marcador 3D sobre dichas vistas. Sobre cada par de cámaras se toma aquella asociación que posea la menor distancia y contenga puntos válidos, descartando las restantes.

Algoritmo: *Asociar puntos 2D.*

$$\begin{array}{l} \forall \text{ cámara } i \text{ y } j \neq i \\ \quad \forall \text{ punto } x_{in} \text{ en cámara } i \\ \quad \quad \text{Hallar recta epipolar } l_{in} \text{ en cámara } j \\ \quad \quad \forall \text{ punto } x_{jm} \text{ en cámara } j \\ \quad \quad \quad \lfloor b_{nm} = \text{distancia}(x_{jm}, l_{in}) \\ \quad \quad \quad \lfloor D_{ij} = \{b_{nm}\}, \text{ posibles asociaciones} \end{array}$$

Algoritmo: *Mejor asociación.*

$$\begin{array}{l} \forall \text{ cámara } i \text{ y } j \neq i \\ \quad \tilde{D}_{ij} = \text{válidos}(D_{ij}) \\ \quad b_{nm}^{ij} = \text{mínimo}(\tilde{D}_{ij}) \\ \quad x_{in}x_{jm} \text{ pareja asociada a } b_{nm}^{ij} \\ \quad d_{ij} = \text{distancia entre rayos de } x_{in} \text{ e } x_{jn} \\ \quad d = \text{mínimo}(\{d_{ij}\}) \end{array}$$

Es importante resaltar que para la elección de los pares de cámaras se han considerado dos casos. El primero de ellos evalúa cada cámara respecto a todas las restantes y el segundo considera la disposición de las cámaras en el espacio y empareja las cámaras adyacentes de manera consecutiva.

Reconstrucción 3D y validación. La pareja de puntos $x_i x_j$ de las cámaras i y j respectivamente, reconstruyen un punto 3D X_{ij} válido si existe al menos un x_k en cámara $k \neq i, j$ tal que $X_{ik} \in \text{Bola}(X_{ij}, \delta)$, para cierto umbral δ .

Actualizar asociaciones. Se retira la pareja que genera la reconstrucción X_{ij} así como también a los puntos x_k que lograron validarla y se itera nuevamente repitiendo el proceso con la siguiente mejor pareja asociada entre dos cámaras. Finalmente el proceso iterativo se detiene cuando el número de marcadores reconstruidos es igual al número de marcadores que tiene colocada la persona, igual al número máximo de marcadores reconstruidos que se haya indicado, o no existen puntos 2D válidos tal que pueda establecerse una asociación entre puntos de distintas vistas.

3.4. Seguimiento

El seguimiento de trayectorias se realiza sobre una ventana deslizante de tres a cuatro cuadros enlazando los puntos reconstruidos de manera de mantener un movimiento lo mas suave posible. Esta metodología fue utilizada por Herda [1] en su trabajo basándose en los estudios de Malik, Drako, Papantoniou [15] .

Algoritmo. Sea la trayectoria de un marcador enlazada hasta el instante [f] sobre la cual desea buscarse su próximo punto en [f+1], el movimiento entre [f-1] y [f] es prolongado para establecer un centro de búsqueda y encontrar el punto reconstruido que mejor continúa la trayectoria como se muestra en la Figura 4 .

Se presentan tres posibles casos al buscar puntos reconstruidos:

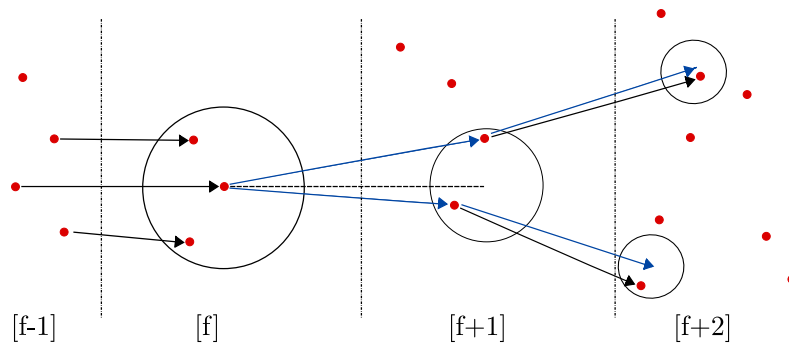


Figura 4: Seguimiento en cuatro cuadros, siendo $[f]$ el cuadro actual que queremos seguir en $[f+1]$. (Fuente: Herda L, Fua P, Plänklers R, Boulic R and Thalmann D, 2011 *Human movement science* 20(3), p.319).

- Si solo se encuentra un punto reconstruido se agrega a la trayectoria para el cuadro $[f+1]$, buscando el mas cercano a la estimación calculada como aquella que mejor se aproxima a una trayectoria de tres puntos con aceleración mínima
- En el caso de encontrar mas de un punto cada posible candidato es evaluado para realizar una segunda estimación hacia $[f+2]$ de forma que la aceleración entre $[f-1]$, $[f]$ y el candidato en $[f+1]$ sea la misma que entre $[f]$, el candidato en $[f+1]$ y la estimación en $[f+2]$. Luego de todos los posible caminos en cuatro cuadros, se elige el de menor variación de aceleración.
- Si no se encuentra ningún punto, se procede a aumentar de forma limitada el radio de búsqueda en $[f+1]$ de forma excepcional. Esto se hace para continuar trayectorias que entran en estado de reposo y el último movimiento conocido es nulo o muy pequeño.

Si una trayectoria queda trunca durante el enlace, se intenta recuperar prolongando el movimiento en próximas cuadros para encontrar puntos reconstruidos cercanos a las estimaciones y extrapolar los puntos intermedios. Por otro lado, se implementan umbrales para definir límites sobre la aceleración de los enlaces obtenidos y detectar discontinuidades durante el seguimiento.

Estas medidas permiten detectar trayectorias individuales sobre los puntos reconstruidos, detectar de forma simple posibles discontinuidades, y estimar reemplazos en casos de pérdidas. En la Figura 5 se muestra el resultado de procesar imágenes de una secuencia real de marcha con zancada exagerada de un hombre sano, provenientes de 8 cámaras del laboratorio virtual que mantienen una visión doble desde cada una de las esquinas de un espacio de captura rectangular. Se resaltan las trayectorias individuales de tres puntos 9 – 10 – 12 correspondientes a pie, rodilla y parte superior de la pierna derecha, así como un esqueleto simple generado para visualizar la evolución entre marcadores.

El conjunto de puntos reconstruidos puede ser sometido a otros algoritmos de seguimiento como Kalman [16] requiriendo la inicialización de modelos, o algoritmos basados en restricciones más fuertes que utilicen las distancias relativamente constantes entre marcadores de los miembros y ángulos continuos entre articulaciones, requiriendo un mayor estudio de las características del sujeto y movimiento a capturar.

4. Resultados

Fueron utilizadas las métricas establecidas en HumanEva [17] para comparar los conjuntos de datos obtenidos a la salida de cada bloque individual con aquel conjunto de referencia obtenido en el ground truth de base de datos, buscando primero la correspondencia entre puntos y luego

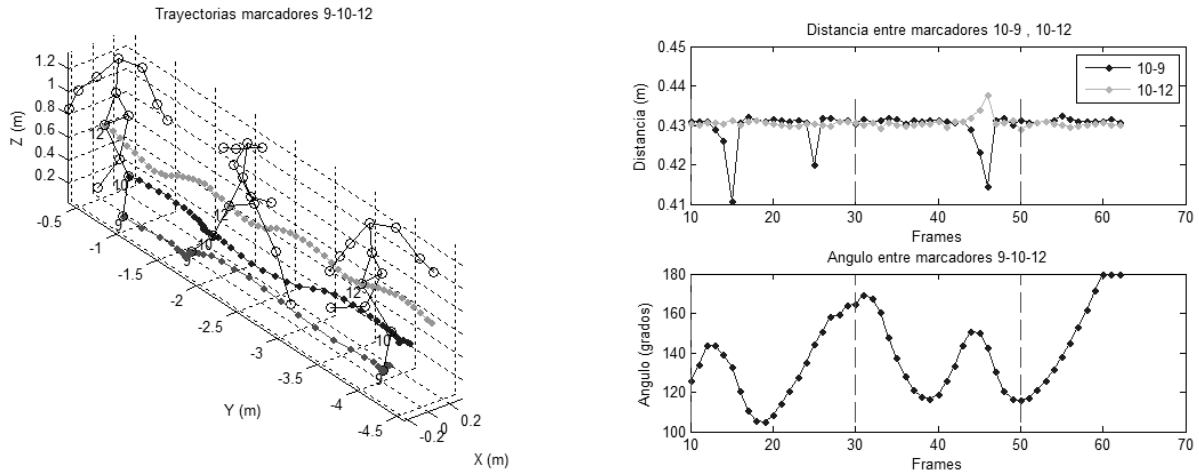


Figura 5: Posibles restricciones en ángulo y distancia, para el caso de la pierna en marcha. Izquierda: trayectorias de marcadores de pierna. Derecha: distancia y ángulo entre marcadores de la pierna.

la distancia euclidiana (en 2D para cámaras y 3D para espacio reconstruido) entre los puntos de ambos conjuntos.

El error de detección de los marcadores en cada una de las cámaras no supera un par de píxeles para el caso de cámaras con resolución en imagen de 1600×600 . Es posible reducir la resolución de las cámaras hasta 800×300 manteniendo los mismos resultados, pero en resoluciones menores comienza a degradarse la tasa de detección de marcadores en cámaras individuales lo cual perjudica las siguientes etapas. Inyectando ruido en el bloque de detección se cuantificó su impacto en etapas posteriores encontrando que se puede trabajar con hasta tres píxeles de error sin comprometer significativamente el error final.

Si las cámaras se encuentran en la condición de error anterior, una cobertura total con 17 cámaras rodeando el espacio de captura permite reconstruir todas las trayectorias con un error medio por debajo del centímetro con errores máximos cercanos a tres centímetros. Los resultados se mantienen dentro de estos límites utilizando ocho cámaras, un par en cada esquina del espacio de captura. La documentación del proyecto [8] muestra como bajo ciertas condiciones se puede seguir reduciendo el número de cámaras manteniendo una performance aceptable.

Acción	Nombre Secuencia	Longitud de secuencia (segundos)	Cuadros por segundo	Espacio de captura (metros)
Marcha rectilínea	08_03_100_100	2,5	30	1,5 x 5,5
Marcha rectilínea, zancada exagerada	08_07_100_100	2,6	24	1,5 x 5,5
	08_07_100_200		48	
Marcha rectilínea lenta, zancada ancha	08_11_100_100	3,8	30	1,5 x 5
Corriendo	09_07_100_100	1,2	24	1,5 x 5
	09_07_100_200		48	
Marcha libre	09_12_100_100	12,5	24	3 x 5

Cuadro 1: Resumen de secuencias de la base de datos sintética utilizadas para análisis de performance.

captura	markers	cuadros	n.cams	Segmentación		Reconstrucción		Seguimiento	
				Promedio (px)	99% (px)	Promedio (cm)	99% (cm)	Promedio (cm)	99% (cm)
08_03_100_100	14	89	17	1,11	3,68	0,41	2,64	0,42	2,90
08_07_100_100	14	62	17	1,09	3,12	0,34	1,81	0,34	1,81
08_07_100_200	14	123	17	1,09	3,31	0,39	1,27	0,36	1,27
08_11_100_100	13	94	17	1,07	2,90	0,39	2,87	0,39	2,87
09_07_100_100	14	29	17	1,13	3,32	0,28	2,04	0,28	2,04
09_07_100_200	14	57	17	1,14	3,45	0,35	1,99	0,35	1,99
09_12_100_100	13	300	15	1,04	2,12	0,40	1,50	0,40	1,50

Cuadro 2: Resultados de las etapas de procesamiento, para distintas capturas sintéticas

5. Conclusiones

Se obtuvo en forma íntegra un sistema de procesamiento para el análisis de capturas de movimiento basadas en marcadores, que a partir de las capturas de video de una persona en un ambiente de laboratorio con las condiciones adecuadas, obtiene la posición 3D de los marcadores presentes en el cuerpo de dicha persona, logrando representar su movimiento con errores medios por debajo del centímetro.

Por otro lado, la implementación separa cada etapa del proceso en módulos distintos, capaces de funcionar de manera independiente. Lo cual permite que el sistema no sea estrictamente óptico sino lo bastante general como para funcionar con cualquier sistema de adquisición que genere imágenes.

Al realizar pruebas con secuencias reales provenientes de tres cámaras en un laboratorio fuera de las hipótesis de captura, se producen problemas en las etapas de segmentación y reconstrucción. Efectuando extracción de fondo y modificaciones sobre la reconstrucción se logran mejores resultados aunque apenas aceptables. Constatando el fuerte impacto que tiene una pobre metodología de captura sobre el posterior procesamiento.

Se desarrolló además un laboratorio virtual de captura de movimiento, permitiendo por un lado, modificar fácilmente las condiciones de captura (número de cámaras, iluminación, etc.) y por otro tener el ground truth de las secuencias generadas. Esto permite tener una plataforma flexible sobre la cual realizar tests de performance y benchmarking sobre cualquier conjunto de algoritmos de captura de movimiento.

Se logra entonces contribuir con reproductibilidad y metodología de diseño en el área a través de un sistema de procesamiento completo y estructurado para la captura de movimiento.

Referencias

- [1] Herda L, Fua P, Plänkner R, Boulic R and Thalmann D 2001 *Human movement science* **20** 313–341
- [2] Vicon <http://www.vicon.com> accedido 30-11-2014
- [3] Optitrack <http://www.naturalpoint.com/optitrack/> accedido 30-11-2014
- [4] Phasespace systems <http://www.phasespace.com/index.html> accedido 30-11-2014
- [5] Qualisys <http://www.qualisys.com/> accedido 30-11-2014
- [6] Motion analysis <http://www.motionanalysis.com/index.html> accedido 30-11-2014
- [7] Kinovea <http://www.kinovea.org/> accedido 30-11-2014
- [8] Guchin A, Pereira G, Ottado G and Ramos M 2015 *Análisis de video en Biomecánica* Proyecto fin de carrera Universidad de la República URL <https://drive.google.com/file/d/0B7mu6FFJvDaHNXhwS2pVQmFHSkE/view?usp=sharing>
- [9] cgspeed <http://www.cgspeed.com/> accedido 6-12-2014
- [10] Warren M, McKinnon D and Upcroft B 2013 *International Conference on Robotics and Automation (ICRA)* (Karlsruhe)
- [11] Medioni G and Kang S B 2004 *Emerging topics in computer vision* (Prentice Hall PTR)
- [12] Svoboda T, Martinec D and Pajdla T 2005 *PRESENCE: Teleoperators and Virtual Environments* **14** 407–422 ISSN 1054-7460

- [13] Otsu N 1975 *Automatica* **11** 23–27
- [14] Hu M K 1962 *Information Theory, IRE Transactions on* **8** 179–187
- [15] Malik N, Dracos T and Papantoniou D 1993 *Experiments in Fluids* **15** 297–294
- [16] Kalman R E 1960 *Journal of Fluids Engineering* **82** 35–45
- [17] Sigal L, Balan A O and Black M J 2010 *International journal of computer vision* **87** 4–27