

Predicción de trayectorias usando el filtro de Kalman

Astrid C. Estrada-Trejo, Isaac Ramos-Silva, Gabriel Sepúlveda-Cervantes,
Edgar A. Portilla-Flores, Eduardo Vega-Alvarado

Instituto Politécnico Nacional,
Centro de Innovación y Desarrollo Tecnológico en Cómputo,
Ciudad de México, México
email{astrid.adartse.10,isaac92ramos}@gmail.com,
{gsepulvedac,aportilla,evega}@ipn.mx

Resumen. El uso de las tecnologías de cómputo ha tenido un gran auge en los últimos veinte años, aumentando la capacidad de los humanos para interactuar con ambientes que antes parecía imposible recrear. De esta manera, el usuario puede ser capaz de percibir el mundo real a través de la virtualización del mismo, dando pie a que tenga una experiencia sensorial brindada por la computadora. En este trabajo se integra un sistema de captura de movimiento en 3D con un entorno virtual para permitir al usuario interactuar entre el entorno físico y el entorno virtual, aprovechando el medio virtual para desarrollar una interacción atemporal entre el usuario y el entorno real. Por medio de la computadora se presentan los cálculos de predicción de posición dado un objeto a analizar, para que el usuario visualice el comportamiento del objeto a lo largo del tiempo mientras esté inmerso en el entorno virtual, lo que le permite interactuar con el entorno real de diferente manera.

Palabras clave: predicción de trayectoria, filtro de Kalman, realidad virtual.

Trajectory Prediction Using Kalman Filter

Abstract. The use of computer technologies has been booming in the last twenty years, increasing the ability of humans to interact with environments that previously seemed impossible to recreate. In this way, the user can be able to perceive the real world through the virtualization of the same, giving rise to having a sensory experience provided by the computer. This work integrates a 3D motion capture system with a virtual environment to allow the user to interact between the physical environment and the virtual environment, taking advantage of the virtual

environment to develop a timeless interaction between the user and the real environment. The calculations of position prediction given by an object to be analyzed are presented by means of the computer, so that the user can visualize the behavior of the object over time while immersed in the virtual environment, which allows it to interact with the real environment in different ways.

Keywords: Trajectory prediction, Kalman filter, virtual reality.

1. Introducción

La realidad virtual (RV) es una herramienta que ofrece una amplia gama de posibilidades para presentar información, de tal manera que el usuario o receptor la perciba e intérprete en manifestaciones diferentes a las imágenes planas o las nomenclaturas y números, incluyendo además una experiencia sensorial proporcionada por una computadora. En este trabajo se utiliza la RV para la predicción de trayectoria de un objeto rígido con movimiento parabólico, con la intención de que el usuario pueda interactuar con dicho objeto en la posición final calculada.

Predecir se define como *anunciar por revelación, ciencia o conjetura algo que ha de suceder* [1]. Con respecto a las trayectorias, el término predicción se refiere a la aproximación al estado actual calculada a partir de un conjunto de variables de salida de estados (posiciones) anteriores. Existen diferentes métodos de predicción dependiendo del modelo a predecir: basado en leyes, de caja negra o sin modelo explícito [2]. En este trabajo, con el fin de anticipar la posición de un objeto que cumple la trayectoria de una parábola se aplicó un estimador de estados como método de predicción. El filtro de Kalman es un conjunto de ecuaciones matemáticas que implementan un estimador óptimo del tipo predictor - corrector, que procesa todas las medidas disponibles, sin importar su precisión, para estimar el valor actual de las variables de interés [3].

Para el cálculo de trayectorias se han desarrollado diferentes enfoques. En [4], se propone un método para integrar un algoritmo en un lazo abierto que genera una guía de trayectoria hacia el punto de impacto predicho. Para ello se considera la dinámica de tres cuadrirrotores y la posición y radio de una red, ya que se plantea lanzar y atrapar una pelota con dicha red sostenida por un sistema de tres drones. El algoritmo de predicción se basa en la integración de estimaciones de velocidad y posición actuales mediante un filtro de Kalman. Por su parte, en [5] se presenta un sistema de percepción que rastrea y predice la posición de dos pelotas en movimiento con ayuda de cámaras posicionadas en la cabeza de un robot humanoide, el cual tiene la tarea de atraparlas. El sistema muestra las trayectorias pronosticadas de las dos bolas lanzadas simultáneamente, así como un horizonte artificial que representa la orientación de la cámara. La estimación de la posición y la velocidad de la pelota se realiza mediante filtros de Kalman extendidos (*Extended Kalman Filters*, EKF), o en ciertos casos ajustando una parábola a las mediciones de las cámaras.

Uno de los trabajos más destacados en el área contempla el uso de la realidad virtual para atrapar una pelota real utilizando un sistema de captura para rastrear su movimiento, considerando la ubicación de las manos y la cabeza del receptor [6]. La escena se *renderiza* y se proyecta a través de un casco de realidad virtual. Se desarrollan tres visualizaciones: la primera es la representación de una pelota virtual coincidente con la real, la segunda muestra la trayectoria pronosticada de la pelota y un punto final, y la última únicamente incluye el punto final. Usando filtros de Kalman tipo *Unscented* se generan estimaciones predictivas del movimiento de la pelota a medida que se acerca al receptor.

En el presente trabajo se desarrolla la integración de un sistema de captura de movimiento en 3D con un entorno virtual, que permite al usuario interactuar tanto con el entorno físico como con el virtual, siendo uno reflejo del otro. Así, se aprovecha el entorno virtual para desarrollar una interacción atemporal entre el usuario y el entorno real. Por medio de la computadora se realizan los cálculos de predicción dado un objeto a analizar, en este caso una pelota, la cual será lanzada al usuario para que este la atrape teniendo la oportunidad de visualizar su comportamiento a lo largo del tiempo incluyendo su posición final antes de que ésta ocurra. El algoritmo de predicción es un estimador de estados basado en el filtro de Kalman, que dadas la posición inicial y la velocidad determina la posición final y la trayectoria del objeto analizado.

El trabajo está organizado de la siguiente manera: en la Sección 2 se desarrolla el modelo propuesto para este proyecto, con el filtro de Kalman, junto a su desarrollo matemático. En la Sección 3 se describe la integración del sistema, el algoritmo computacional y su implementación, mientras que en la Sección 4 se presenta la experimentación y sus resultados. Finalmente, las conclusiones y trabajos a futuro se incluyen en la Sección 5.

2. Modelo propuesto

2.1. Tiro parabólico

El tipo de trayectoria predeterminado es el tiro parabólico, por lo que se trabajará su ecuación cinemática mediante los métodos de integración de Euler.[7] Para ello, se proponen los vectores x , y , z , dependientes del tiempo, tales que su descomposición (individual) implica una ecuación cuya diferencial se relaciona con la posición respecto a su primer instante en un espacio $(t-1)$. Dichos vectores se indican en las ecuaciones (1) a (3):

$$x_{(t)} = x_{(t-1)} + \dot{x}_{(t)}\Delta_{(t)}, \quad (1)$$

$$y_{(t)} = y_{(t-1)} + \dot{y}_{(t)}\Delta_{(t)}, \quad (2)$$

$$z_{(t)} = z_{(t-1)} + \dot{z}_{(t)}\Delta_{(t)}. \quad (3)$$

Siguiendo el método de integración de Euler, se aplica la segunda derivada para que la diferencia de posición se convierta en una diferencia de velocidad con respecto al tiempo. Esto se muestra en las ecuaciones (4) a (6):

$$\dot{x}_{(t)} = \dot{x}_{(t-1)} + \ddot{x}\Delta_{(t)}, \quad (4)$$

$$\dot{y}_{(t)} = \dot{y}_{(t-1)} + \ddot{y}\Delta_{(t)}, \quad (5)$$

$$\dot{z}_{(t)} = \dot{z}_{(t-1)} + \ddot{z}\Delta_{(t)}. \quad (6)$$

La variable \ddot{y} es la aceleración debido a la fuerza de gravedad. Dado que su valor es despreciable en las dimensiones x y z , queda expresada solo en y como $\ddot{y} = -9,81 \text{ m/s}^2$. Las ecuaciones (7) a (9) corresponden a los vectores de estado expresados en forma matricial,

$$x_{(t)} = \begin{bmatrix} x \\ \dot{x} \end{bmatrix} = \begin{bmatrix} x + \Delta_{(t)}\dot{x} \\ \dot{x} \end{bmatrix}, \quad (7)$$

$$y_{(t)} = \begin{bmatrix} y \\ \dot{y} \end{bmatrix} = \begin{bmatrix} y + \Delta_{(t)}\dot{y} \\ \dot{y} + \Delta_{(t)}\ddot{y} \end{bmatrix}, \quad (8)$$

$$z_{(t)} = \begin{bmatrix} z \\ \dot{z} \end{bmatrix} = \begin{bmatrix} z + \Delta_{(t)}\dot{z} \\ \dot{z} \end{bmatrix}. \quad (9)$$

2.2. Predicción de la trayectoria

Aplicando el modelo del filtro de Kalman *unscented* considerado para un caso discreto de estimación de estados y realizado para cada vector por separado, se tienen modelos de 1D [3]. La ecuación convencional de la proyección del estado hacia adelante se indica en la expresión (10) [8]:

$$x_{(t)} = Ax_{(t-1)} + Bu_{(t)}. \quad (10)$$

Así, se puede considerar una función de transición g la cual, dadas las variables de posición $x_{(t)}$, aceleración \ddot{y} y un paso de tiempo $\Delta_{(t)}$ regresa una aproximación de la posición siguiente $x_{(t+1)}$. Las ecuaciones (11), (12) y (13) corresponden a tal función en x , y y z , respectivamente, con $x_{(t)}, g(x_{(t)}, \ddot{y}, \Delta_{(t)}) \in \mathbb{R}^2$:

$$g(x_{(t)}, 0, \Delta_{(t)}) = \begin{bmatrix} x + \Delta_{(t)}\dot{x} \\ \dot{x} \end{bmatrix} = \begin{bmatrix} 1 & \Delta_{(t)} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \end{bmatrix}, \quad (11)$$

$$g(y_{(t)}, \ddot{y}, \Delta_{(t)}) = \begin{bmatrix} y + \Delta_{(t)}\dot{y} \\ \dot{y} + \Delta_{(t)}\ddot{y} \end{bmatrix} = \begin{bmatrix} 1 & \Delta_{(t)} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} y \\ \dot{y} \end{bmatrix} + \begin{bmatrix} 0 \\ \Delta_{(t)} \end{bmatrix} \ddot{y}, \quad (12)$$

$$g(z_{(t)}, 0, \Delta_{(t)}) = \begin{bmatrix} z + \Delta_{(t)}\dot{z} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} 1 & \Delta_{(t)} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} z \\ \dot{z} \end{bmatrix}. \quad (13)$$

Entonces, $g'(x_{(t)}, \ddot{y}, \Delta_{(t)})$ es una matriz de 2x2 definida como la derivada parcial de g con respecto a $x_{(t)}$ para cada uno de sus componentes. En otras palabras, se considera como la traspuesta de la matriz A de la ecuación (10), que es la covarianza del error de la función de transición $G_{(t)}$. La ecuación (14) corresponde a dicha función:

$$G_{(t)} = g'(x_{(t)}, \ddot{y}, \Delta_{(t)}) = \begin{bmatrix} 1 & 0 \\ \Delta_{(t)} & 1 \end{bmatrix}. \quad (14)$$

Para la distribución del error ($\bar{\Sigma}_{(t)}$) se supone un error inicial de 1 en la distribución del error pasado $\Sigma_{(t-1)}$. De esta manera, la función dispone de un ruido del proceso $Q_{(t)}$, una covarianza del error de la función de transición $G_{(t)}$ y la transpuesta de la covarianza del error de la función de transición $G_{(t)}^T$. Así, se tiene la ecuación (15), sujeta a las observaciones calculadas por la ecuación convencional de observación, expresión (16),

$$\bar{\Sigma}_{(t)} = G_{(t)}\Sigma_{(t-1)}G_{(t)}^T + Q_{(t)}, \tag{15}$$

$$y_{(t)} = Cx_{(t)}. \tag{16}$$

Expresando la ecuación (16) como la función de observación $h(x_{(t)})$ dada en el vector de la posición x , y o z , y entregando la lectura con mayor probabilidad dado el estado actual se obtienen las ecuaciones (17) a (20):

$$h(x_{(t)}) = [H, x_{(t)}], \tag{17}$$

$$h(x_{(t)}) = [1 \ 0] \begin{bmatrix} x \\ \dot{x} \end{bmatrix}, \tag{18}$$

$$h(y_{(t)}) = [1 \ 0] \begin{bmatrix} y \\ \dot{y} \end{bmatrix}, \tag{19}$$

$$h(z_{(t)}) = [1 \ 0] \begin{bmatrix} z \\ \dot{z} \end{bmatrix}. \tag{20}$$

Por las características del Jacobiano, se define la función $h'(x_{(t)})$ de la ecuación (21) como una matriz de 1x2, para calcular la ganancia de Kalman,

$$h'(x_{(t)}) = [\Delta_{(t)}, (\Delta_{(t)})'] . \tag{21}$$

Puesto que es un sistema lineal, H' es igual a H , donde:

$$h'(x_{(t)}) = [1 \ 0] . \tag{22}$$

Es necesario calcular los reajustes de la covarianza de la predicción y la suma de las posibles mediciones con respecto al tiempo, utilizando la ecuación 23 de corrección, donde $K_{(t)}$ es la ganancia de Kalman que minimiza los reajustes.

$$K_{(t)} = \bar{\Sigma}_{(t)}H_t^T(H_t\bar{\Sigma}_{(t)}H_t^T + R_{(t)})^{-1}. \tag{23}$$

A continuación se actualizan el valor estimado de la posición y su covarianza de error siendo ahora $\mu_{(t)}$ la nueva posición con una covarianza reajustada $\Sigma_{(t)}$ que da cálculo al posible error ($\bar{\Sigma}_{(t)}$), empleando las ecuaciones (24) y (25):

$$\mu_{(t)} = \bar{\mu}_{(t)} + K_{(t)}(z_{(t)} - h(\bar{\mu}_{(t)})), \tag{24}$$

$$\Sigma_{(t)} = (I - K_t H_t)\bar{\Sigma}_{(t)}. \tag{25}$$

3. Integración

3.1. Sistema de captura de movimiento

La captura de movimiento o *motion capture* (mocap) es el proceso de registrar el movimiento de objetos o personas. En este caso se usará la técnica de captura tipo óptico-pasivo, la cual usa marcadores reflectantes que son rastreados por cámaras infrarrojas [9]. Este proceso se muestra en la Figura 1b, con el sistema Vicon utilizado en este desarrollo. Vicon cuenta con el programa *Tracker* (Figura 1a), para visualizar la imagen capturada por las cámaras. Adicionalmente tiene la opción de desplegar imágenes en un plano 3D facilitando la tarea de localización dentro del área. Destaca el software de apoyo para el sistema de captura, ya que permite la interacción con sus elementos (cámaras) y procesa los datos en 2D y 3D.

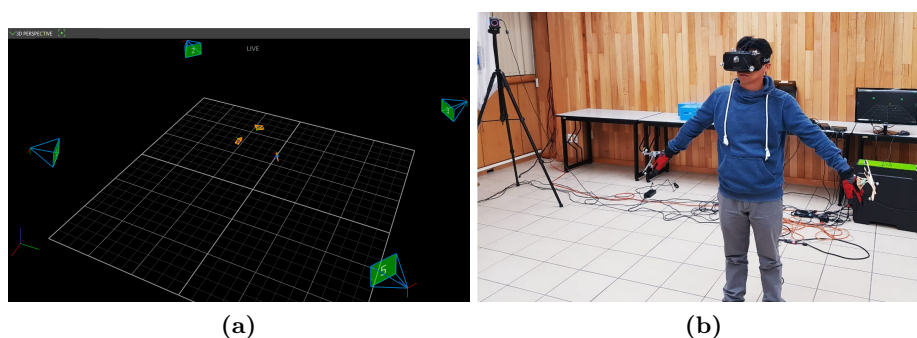


Fig. 1. Sistema Vicon.

3.2. Implementación en Unity

Unity es un motor de videojuegos que permite desarrollar entornos virtuales y visualizarlos en cascos de realidad virtual, tal como Oculus Rift, para dar al usuario una experiencia inmersiva en un mundo virtual. También permite una comunicación con el sistema de captura de movimiento Vicon, dando los datos de posición de la pelota que son las variables de entrada para el Algoritmo 1: de predicción, en el cual se implementa una estructura de filtro de Kalman Extendido [3]. Se realizan los cálculos respectivos mediante un *script*, que es un programa en lenguaje C# o Java asociado a un *GameObject* (cualquier objeto dentro del entorno de Unity, desde modelos 3D hasta la iluminación, cámara, etc), que le proporciona una funcionalidad dada la programación del script [10]. La función *Update* se encarga de la actualización por marco (*frame*) para el objeto, es decir todo el código que se encuentre dentro de esta función es llamado

de forma automática por el sistema cada marco o fps. Si el proyecto se programa a n fps, *Update* se ejecutará n veces cada segundo [10] permitiendo así que la función de predicción se realice de manera continua y con un tiempo corto en la actualización de datos de posición.

Algoritmo 1: Filtro de Kalman extendido.

Require: Predicción $\{\Sigma_{(t-1)}, \mu_{(t-1)}, x_{(t-1)}, \Delta_{(t)}\}$

$$\begin{aligned} \bar{\mu}_{(t)} &= g(x_{(t)}, \mu_{(t-1)}) \\ \bar{G}_{(t)} &= g'(x_{(t)}, \dot{\mu}, \Delta_{(t)}) \\ \bar{\Sigma}_{(t)} &= G_{(t)} \Sigma_{(t-1)} G_{(t)}^T + Q_{(t)} \end{aligned}$$

return $\bar{\mu}_{(t)}, \bar{\Sigma}_{(t)}$

Require: Corrección $\Sigma_{(t)}, \mu_{(t)}, x_{(t)}$

$$\begin{aligned} H_{(t)} &= h'(\mu_{(t)}) \\ K_{(t)} &= \bar{\Sigma}_{(t)} H_{(t)}^T (H_{(t)} \bar{\Sigma}_{(t)} H_{(t)}^T + R_{(t)})^{-1} \\ \mu_{(t)} &= \bar{\mu}_{(t)} + K_{(t)} (z_{(t)} - h(\bar{\mu}_{(t)})) \\ \Sigma_{(t)} &= (I - K_{(t)} H_{(t)}) \bar{\Sigma}_{(t)} \end{aligned}$$

return $\mu_{(t)}, \Sigma_{(t)}$

$$\bar{\mu}_{(t)}, \bar{\Sigma}_{(t)} = \text{Predicción } \{\Sigma_{(t-1)}, \mu_{(t-1)}, x_{(t-1)}, \Delta_{(t)}\}$$

$$x_{(t-1)} = x_{(t)}$$

$$\mu_{(t)}, \Sigma_{(t)} = \Sigma_{(t)}, \mu_{(t)}, x_{(t)}$$

4. Experimento y resultados

Para las pruebas del sistema se implementó un área de captura de 7.64 m de largo por 5.96 m de ancho, ubicando al usuario en el centro de este espacio tal como se muestra en la Figura 2A. El sujeto de prueba utiliza el casco de realidad virtual que tiene colocados cinco marcadores reflectantes para registrar la posición y altura del usuario. El objeto de interacción tanto en el entorno virtual como en el real es una pelota con cuatro marcadores reflectantes, como se muestra en la Figura 3. La importancia de llevar acabo su registro de posición es por dos razones; la primera es para emparentarla con el modelo 3D de forma que su posición tanto en el entorno real como en el virtual serán los mismos. La segunda razón es porque la posición del objeto es la analizada para calcular por medio del algoritmo de Kalman su trayectoria futura. Estos dos puntos se ven reflejados en la Figura 2B.

Una vez instalado el usuario dentro del área de captura se coloca frente a él un evaluador a metro y medio de distancia. En la primera etapa de interacción el evaluador sostiene la pelota cuestionando al usuario sobre la posición de está al cambiar la distancia y posición de dicha pelota dentro del área de captura. El usuario inmerso en una escena virtual del área de captura real visualiza una pelota virtual, pero al interactuar con ella lo hace de igual manera con la pelota real (véase la Figura 5). En la segunda etapa, el evaluador permanece a una distancia de metro y medio frente al usuario ahora con la intención de lanzar

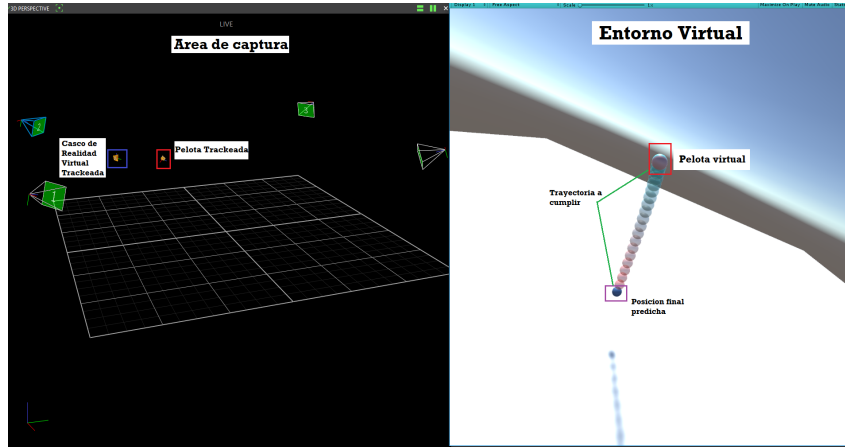


Fig. 2. A) Área de captura donde el usuario interactúa con el objeto real. B) Representación en el entorno virtual, con la trayectoria y la posición final.



Fig. 3. Elementos del sistema de captura de movimiento.

la pelota y que el sujeto de estudio la atrape. En la tercera etapa se activa la función de predicción y se repite la segunda etapa, realizando un intercambio de lanzamientos de la pelota con la ventaja para el usuario al visualizar de manera desplegada la traslación de su posición y su punto final antes de que ocurran.



Fig. 4. A) Interacción con la pelota sin mostrar la predicción. B) Interacción con la pelota visualizando la predicción.

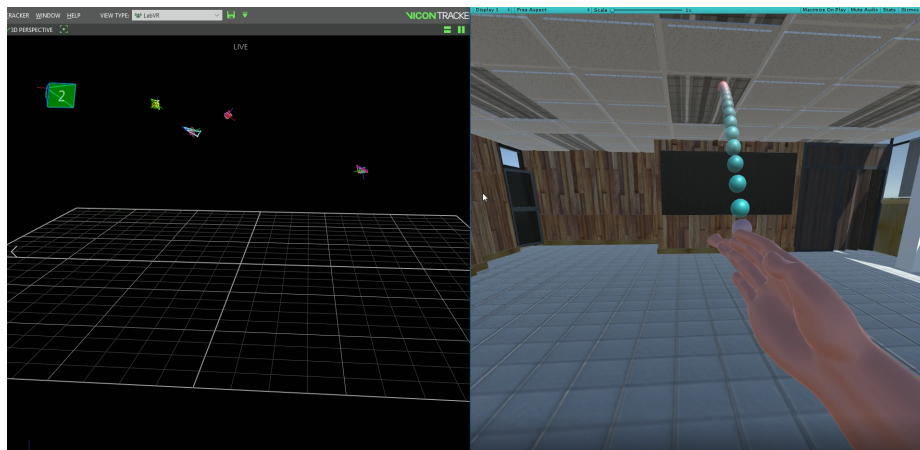


Fig. 5. Entorno virtual del área de captura, donde el usuario tiene además una visualización de una mano 3D que sigue los movimientos de su propia mano.

5. Conclusiones y trabajos a futuro

Es evidente el cambio de actitud en el usuario al visualizar la trayectoria que esta cumpliendo la pelota y su punto final, modificando su postura para atraparla como se observa en la figura 4. La interacción atemporal se da de manera satisfactoria al visualizar la predicción la posición gracias a la integración de los cálculos matemáticos del filtro de Kalman *unscented* en un algoritmo de filtro de Kalman extendido, ya que varias de las matrices se convierten en constantes que no se requiere recalcular.

Una vez completado este proyecto, como trabajo a futuro se tiene planeado desarrollar estudios donde la variable es el entorno de interacción del usuario, para determinar en cual tiene un mayor rendimiento el uso de la predicción, ya sea el entorno real, la realidad virtual, la realidad aumentada o alguna otra

variación entre estas. Para ello se requiere integrar modelos 3D que permitan mayor inmersión en el entorno virtual, por ejemplo manos para el usuario, algún indicador de la posición del evaluador, etc., tal como se muestra en la figura 5.

Agradecimientos. Los autores agradecen al Instituto Politécnico Nacional por el apoyo recibido a través de la Secretaría de Investigación y Posgrado por medio de los proyectos SIP20196288, SIP20190245 y SIP20195772. El primer autor agradece al CONACyT el apoyo otorgado a través de una beca para estudios de maestría.

Referencias

1. Diccionario de la lengua española. 23^a Edición, Real Academia Española (2014)
2. Arahál, M.R., Soria, M.B., Díaz, F.R.: Técnicas de predicción con aplicaciones en Ingeniería. Vol. 15, Universidad de Sevilla (2006)
3. Tellex, S., Brown, A., Lupashin, S.: Estimation for Quadrotors. arXiv preprint arXiv:1809.00037 (2018)
4. Ritz, R., Müller, M.W., Hehn, M., D'Andrea, R.: Cooperative quadcopter ball throwing and catching. In: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 4972–4978 (2012)
5. Birbach, O., Frese, U., Bäuml, B.: Realtime perception for catching a flying ball with a mobile humanoid. In: 2011 IEEE International Conference on Robotics and Automation, pp. 5955–5962 (2011)
6. Niemeyer, G., Matthew K.X.J.: Pan, Catching a real ball in virtual reality. In: IEEE Virtual Reality (VR), Los Angeles, CA, USA, Marzo 18-22, pp. 269–270 (2017)
7. Hernández, A.G.: Comparación de métodos analíticos y numéricos para la solución del lanzamiento vertical de una bola en el aire. *Latin-American Journal of Physics Education*, 2(2), 13 (2008)
8. Díaz, J.G., Mejías, A.M., Arteaga, F.: Aplicación de los Filtros de Kalman a Sistemas de Control. *Revista INGENIERÍA UC*, 8(1) (2001)
9. Vicon Motion Systems Ltd: What is motion capture? Oxford, Reino Unido. Consulta online diciembre 2017. Disponible en <https://www.vicon.com/what-is-motion-capture>
10. Unity Documentation: <https://docs.unity3d.com/es/current/Manual/index.html>. Último acceso: 2019/03