

Teleoperación de un Servomecanismo a Través de un Mundo Virtual Usando Una Interfaz Gestual¹

Para citar este artículo /
To reference this article /
Para citar este artigo.
Oliveros, A. Diego I.,
& Santamaría, G. Luz
(2014). Teleoperación de un
Servomecanismo a Través
de un Mundo Virtual usando
una Interfaz Gestual. Ingenio
Magno. Vol 5, pp. 72-80

TELE-OPERATION OF A SERVOMECHANISM BY MEANS OF A VIRTUAL WORLD USING A GESTURE-BASED INTERFACE

Diego Iván Oliveros Acosta, M.Sc. (E)
Universidad Santo Tomás. Tunja, Colombia.
e_mail: diegoo.ivan@gmail.com.

Luz Santamaría Granados, M.Sc.
Universidad Santo Tomás. Tunja, Colombia.
e_mail: luz.santamaria@usantoto.edu.co.

Recepción: 2014 - 07 - 01 \ Aceptación: 2014 - 12 - 25

RESUMEN

Este artículo presenta los resultados de investigación del desarrollo de un prototipo que permite el accionamiento real a distancia de un servomecanismo mediante un mundo virtual controlado por una interfaz gestual. La metodología de investigación formula cuatro fases: la interfaz gestual que utiliza los algoritmos de CamShift y MeanShift para la identificación de la orden que se le envía al robot; la comunicación inalámbrica recibe la instrucción de la interfaz gestual para la teleoperación del robot; el mundo virtual que simula el recorrido del avatar en el ambiente lunar, tal como lo realiza el robot en el mundo real; finalmente se plantea la construcción del sistema eléctrico y mecánico del servomecanismo. Este proyecto puede ser referente para el desarrollo de sistemas de teleoperación que permitan la experimentación remota y virtual de desastres naturales o ambientes que generan riesgo para la vida humana.

Palabras Clave: Blender, Interfaz gestual, Mundo virtual, OpenCV y Servomecanismo

ABSTRACT

The teleoperation of robotic devices with control data reduction through immersive virtual worlds using gestural interfaces, seek to reduce the risk of accidents, improve production in human activities and extend the domain of knowledge. This paper presents the research results of the development of a prototype that allows the real remote operation of a servomechanism through a virtual world controlled by a gesture-based human interface. The research methodology is divided into four parts: The Gesture-based human interface that uses CamShift and MeanShift algorithms to identify the order sent to the robot; The wireless communication that receives the instruction from the gestural interface for the teleoperation of the robot; the virtual world simulates the path of the avatar in the lunar environment, just as the robot does in the real world; finally the construction of electrical and mechanical servo system is applied. This project can be a reference for the development of systems that allow remote teleoperation and virtual experiences of natural disasters or environments that are hazardous to human life.

Keywords: Blender, Gesture-based human interface, OpenCV and Servomechanism, Virtual world.

¹ Grupo de Investigación Basado en Redes de Aprendizaje sobre Nuevas Tecnologías (GIDINT). Proyecto de investigación

1. INTRODUCCIÓN

Los sistemas teleoperados son de gran utilidad en la industria, ámbito militar, médico e investigativo, así como también en el mundo comercial, pero requieren gran capacidad de procesamiento debido al manejo de un número significativo de datos, sobretodo en labores de rescate por desastres naturales donde el ser humano no tiene acceso y existe el riesgo de la imprecisión de los sistemas de información. Este es un gran reto para los profesionales en ingeniería que tienen el propósito de integrar tecnologías de visualización 3D en la teleoperación de servomecanismos, para mejorar la interacción entre los sistemas y el hombre, buscando que los dispositivos sean cada vez más ergonómicos, menos invasivos y reduciendo su tiempo de respuesta. Precisamente este trabajo presenta el desarrollo de una interfaz gestual para la teleoperación de un servomecanismo, que el usuario controla interactuando a través de un mundo virtual.

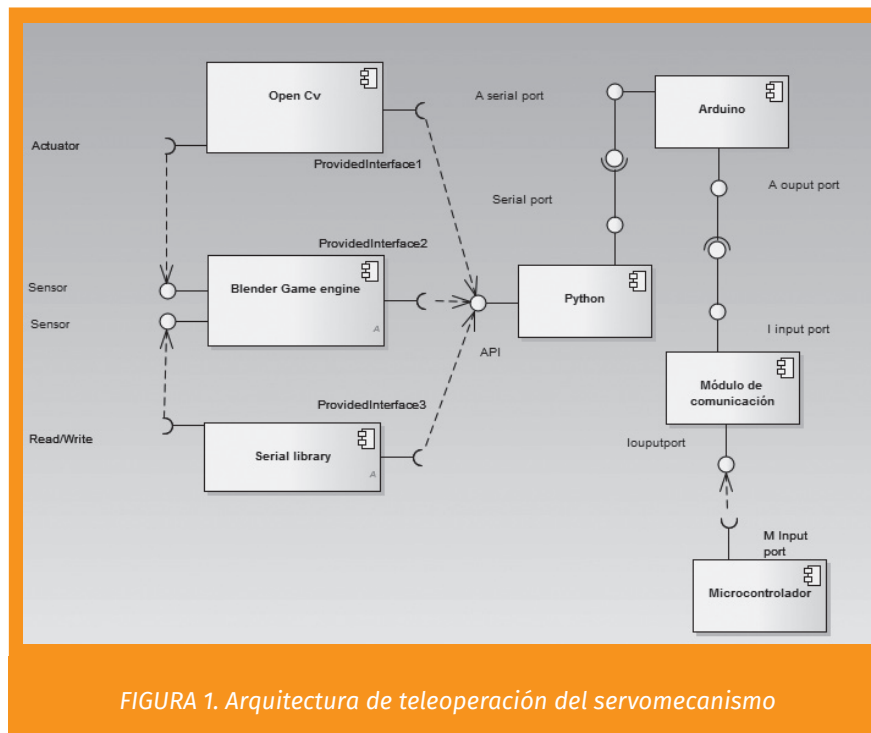


FIGURA 1. Arquitectura de teleoperación del servomecanismo

2. ARQUITECTURA DEL SISTEMA

El diagrama de componentes del sistema de la Interfaz gestual para la teleoperación del servomecanismo a través del mundo virtual se muestra en la figura 1 y está conformado por siete componentes que facilitan la comunicación entre la interfaz gráfica y el robot. El motor de juegos de Blender [1] provee el funcionamiento para que la librería OpenCV [2] haga el reconocimiento de las imágenes, con la librería Serial se procesan los datos recibidos por el usuario desde la interfaz gestual. Como plataforma de soporte para los componentes

iniciales se utiliza el lenguaje de programación Python [3], que facilita la codificación del script que envía los datos a la tarjeta microcontroladora Arduino [4], para que procese la señal de control que envía a través del módulo de comunicaciones al microprocesador, este último funciona en una sola vía para la teleoperación del robot. La arquitectura se considera como cliente / servidor [5], ya que la interfaz gestual se maneja desde el cliente y el control del robot se hace desde el servidor Linux con Python.

3. INTERFAZ GESTUAL

Para el diseño de la interfaz gestual se utiliza el algoritmo de Desplazamiento Medio Continuatmente Adaptativo (Continuously Adaptative Mean Shift Algorithm, CAMSHIFT) [6], que se caracteriza por la rapidez y eficiencia en el seguimiento del movimiento de un objeto que es identificado por la distribución de colores, o cambios de tamaño o de posición. En la figura 2, se muestran los pasos del algoritmo para el seguimiento del objeto (head tracking) de acuerdo a la distribución del color [7]. Inicialmente se escoge la ventana de búsqueda seleccionando el objeto que se quiere seguir, en el proceso de la captura del objeto se hace la conversión de la imagen representada en los tres colores Rojo, Verde y Azul (Red Green Blue, RGB) al modelo de Matiz, Saturación, Valor (Hue, Saturation, Value, HSV), luego se extrae el canal de la matiz y se calcula el histograma de matices que se encuentra en la selección [3]. Para

organizar los colores los valores se escalan en el rango de 0 a 255, posteriormente se obtiene la distribución de probabilidad objetivo. Al determinar la posición del histograma o posición en la que se encuentra el objeto, los ejes verticales son nominados con el prefijo X y los ejes horizontales con el prefijo Y. En cada captura se busca la retroproyección (backprojection) del histograma del paso anterior. Esta operación consiste en generar una imagen en escala de grises, en donde cada pixel tendrá como intensidad el valor del histograma correspondiente a la matriz de dicho pixel en la imagen procesada. El valor de cada pixel de esta imagen determina la probabilidad que el pixel procesado pertenezca al objeto, ya que si el componente de matiz de color es bajo se vuelve poco fiable y no se contabiliza en el histograma [3].

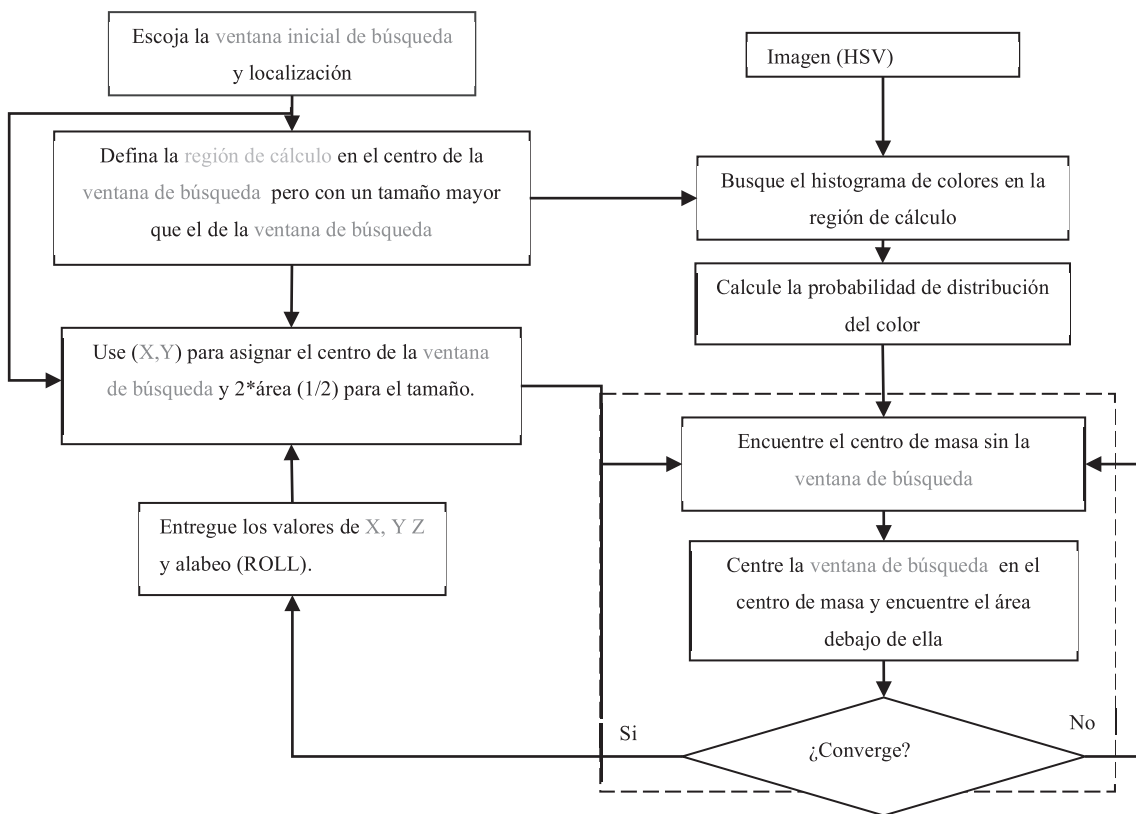


FIGURA 2. Diagrama de seguimiento de color basado en el algoritmo head tracking [2]



Las imágenes se capturan con una resolución de píxeles (640 x 480) y los ejes que definen la matriz de tamaño (3 x 3), definen las posiciones de desplazamiento (izquierda, derecha, adelante y atrás) y los intervalos de control de la interfaz gestual. Se calcula el centroide de la imagen de retroproyección “backprojection” mediante el algoritmo iterativo meanshift [3], busca en un agrupamiento de píxeles la posición del centroide (X, Y) en un punto estable de la ventana predeterminada. Luego, se establece una nueva ventana de búsqueda para las siguientes capturas que utilizan los valores calculados de tamaño, posición del centroide (X, Y) y rotación del objeto en Z (roll). Finalmente se itera el proceso como se muestra en el flujograma de la Figura 2. Python cuenta con bibliotecas gráficas para el modelamiento y renderizado de los objetos 3D [8], tiene soporte a las clases de OpenCV para el análisis de movimiento y seguimiento a los objetos con los algoritmos CamShift y MeanShift [7]. Para el desarrollo del módulo de la interfaz gestual se hizo la modificación a la clase camshift mediante la adición del método MovetoPort, que contiene las coordenadas de la ventana donde se encuentra el nuevo objeto, de acuerdo a este valor este método envía el dato correspondiente al puerto serial para transmitir la instrucción de control al servomecanismo real para que se mueva en las posiciones indicadas. En el módulo de comunicaciones de Radio Frecuencia (Radio Frequency, RF), la librería serial de Blender controla los comandos que se envían por teclado o por la interfaz gestual del usuario, para que sean direccionados desde el puerto serial (COM) a la tarjeta microcontroladora Arduino [4]. La conexión se hace con una interfaz USB, para que desde Arduino sean interpretados los nibbles (4 bits) o la señal de control que el servomecanismo recibirá para moverse de acuerdo a las instrucciones relacionadas en la tabla 1. En la interfaz de Arduino se programaron las funciones que convierten los caracteres a señales digitales.

TABLA 1. Comandos para el robot

No.	Bin	Comando	Motor									
			M1		M2		M3		M4		M5	
			BASE		Rueda Izquierda		Rueda Derecha		Codo		Pinza	
		Polaridad	+	-	+	-	+	-	-	+	DER	IZQ
		PIN PUENTE H	2(1)	7(1)	10(1)	15(1)	2(2)	7(2)	10(2)	15(2)	10(3)	15(3)
		PIN del PIC	35	36	34	33	29	30	28	27	26	25
			B2	B3	B1	B0	D6	D7	D5	D4	C7	C6
1	0001	Adelante			X			X				
2	0010	Atrás				X	X					
3	0011	Derecha			X		X					
4	0100	Izquierda				X		X				
5	0101	Pinza abre									X	
6	0110	Pinza cierra										X
7	0111	Base sube	X									
8	1000	Base Baja		X								
9	1001	Codo sube								X		
10	1010	Codo Baja							X			
11	1011	Adelante con rueda DER						X				
12	1100	Atrás con rueda DER					X					
13	1101	Adelante con rueda IZQ			X							
14	1110	Atrás con rueda IZQ				X						

4. MODELO VIRTUAL

Para la exploración del mundo virtual [9] el usuario cuenta con un avatar [10] que es el modelo del robot en 3D y le permite emular el comportamiento del servomecanismo real. La implementación del algoritmo que controla la interfaz gestual utiliza la librería GameLogic [1], que dispone de un motor gráfico para la programación de sensores, controles y actuadores en el manejo de colisiones,

teclas o acciones del mouse sobre los objetos del ambiente lunar. Blender utiliza procesamiento paralelo para los procesos de la animación, el renderizado y el modelado de los objetos (ver Figura 3). Para el desarrollo del mundo virtual se utilizó la metodología Diseño VRML (Virtual Reality Modelling Language) Efectivo [11], que contempla siete etapas: conceptualización, planificación, diseño, muestreo, construcción, pruebas y publicación.

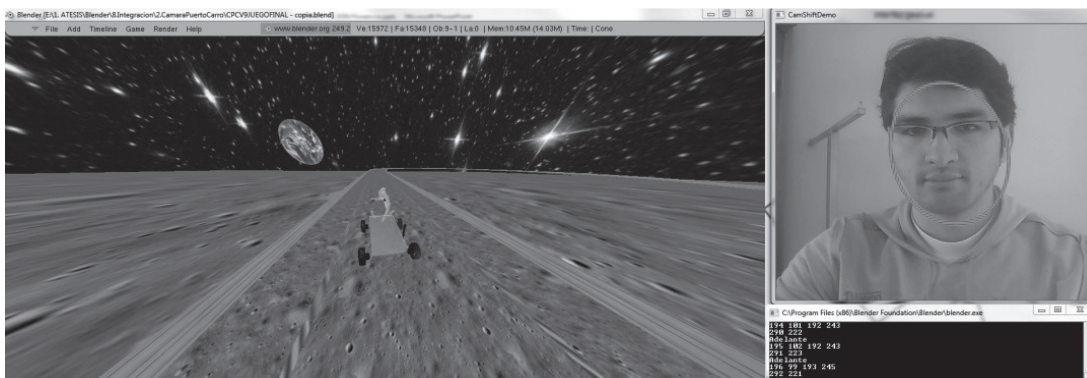


FIGURA 3. Control del videojuego con la interfaz gestual

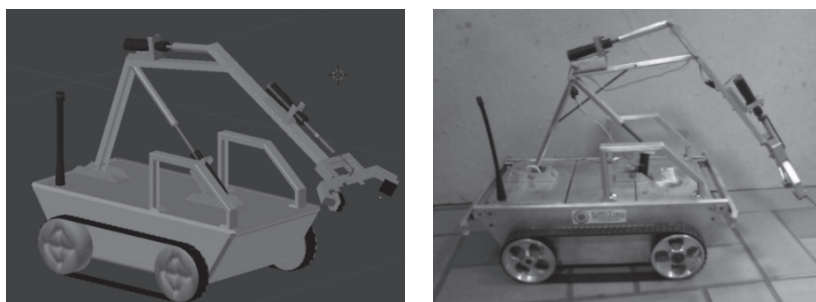


FIGURA 4. Modelo 3D del avatar texturizado y servomecanismo real

En la conceptualización se definió la historia central del videojuego en un entorno ficticio del año 2020, en donde la contaminación y los desastres naturales hicieron de la tierra un completo caos, fue necesario buscar nuevos lugares mediante la exploración espacial. Se construyó un servomecanismo virtual a escala del robot que permitió hacer el recorrido lunar para encontrar organismos vivos. En la etapa de planificación se diseñaron los planos 2D del avatar a escala del robot real, para posteriormente en el diseño adecuar las técnicas de modelamiento para la construcción de los objetos a partir de mallas poligonales. En el proceso de renderizado se usaron los colores del prototipo real (ver figura 4), en lugar de fotografías texturas para mejorar la uniformidad y reducir el tamaño de los archivos del modelo 3D. Una vez creado el avatar se identifica la armadura (rigging), en este paso se analizó la dinámica del servomecanismo teniendo en cuenta los puntos de rotación más no el diseño de la estructura (ver figura 4), el sistema es similar a un juego de pistones. Como técnica de modelamiento se utilizó curve modeling [12] que define una malla detallada de polígonos

triangulares para el refinamiento de los objetos en el escenario lunar. El diseño de las piezas se realizó de manera independiente para poder realizar de forma rápida la asociación de la armadura con los objetos independientes. Para la construcción de las animaciones se programaron los scripts en Python con la interacción de los diagramas de bloques (Logic Bricks). En la etapa de pruebas y de publicación se encontraron problemas de compatibilidad de las versiones de OpenCV y de la librería serial con la versión reciente de Blender (2.61) en la fecha del desarrollo del proyecto, por lo anterior se definió como versión de trabajo Blender (2.49).

5. TELEOPERACIÓN DEL ROBOT

Este módulo se utiliza para controlar el servomecanismo de forma remota y materializar las acciones del robot virtual, con la metodología RCM (Reliability Centered Maintenance) se realizó el mantenimiento preventivo y correctivo del prototipo de robot, construido previamente por el grupo

de investigación GINSCON [13]. En el desarrollo de este proyecto, se adaptó el circuito de control y el circuito de potencia o puente H a este robot. En la tabla 1 se relacionan los puertos que deben ser activados (B0, B1, B2, B3, D4, D5, D6, D7, C6 y C7) cuando la palabra de control o comando (en binario), se ejecute (se presenta en los 4 pines del puerto de entrada del microcontrolador). La implementación del circuito se hizo en el protoboard para hacer el cambio de la lógica del programa del microcontrolador. La tarjeta controla seis motores con la misma distribución de espacio, cuenta con un sistema de alimentación y utiliza el regulador L7805, que puede ser remplazado sin tener un sistema de alimentación independiente para la etapa de control. El pistón transforma la rotación del motor mediante un tubo roscado (espárrago) a desplazamiento lineal por medio de una arandela. La pinza que tiene el mismo sistema con la variante que el desplazamiento lineal de la arandela, se usa para desplazar las palancas que permiten su apertura y cierre. Para cualquiera de los casos es de mayor importancia la velocidad de rotación que el torque. El interior de la plataforma móvil del servomecanismo aloja los motores, circuitos de potencia, comunicación inalámbrica, control y baterías. En el circuito de control se programaron las funciones que recibieron los cuatro bits del módulo de comunicación inalámbrica, para controlar el movimiento de las ruedas y la pinza del robot de acuerdo a las instrucciones generadas desde la interfaz gestual.

TABLA 2. Seguimiento de un objeto por la distribución de colores

Posición	X1	Y1	R1	R2	Coordenada X centro	Coordenada Y centro	Blender	Puerto
Datos para el cálculo	P1*	P2*	Ancho	Alto	$P1 = P1* + \text{ancho}/2$	$P2 = P2* + \text{alto}/2$		
Límite de zonas	A1=0	A2=240	A3=300	A4=640	Y1=0	Y2=50	Y3=80	Y4=480
Adelante izquierda	$P1 \geq 0$	$P1 < 240$	$P2 \geq 0$	$P2 < 50$	$P1 \geq A1$ y $P1 < A2$	$P2 \geq Y1$ y $P2 < Y2$	Q	1101
Izquierda	$P1 \geq 0$	$P1 < 240$	$P2 \geq 50$	$P2 < 80$	$P1 \geq A1$ y $P1 < A2$	$P2 \geq Y2$ y $P2 < Y3$	A	0100
Atrás izquierda	$P1 \geq 0$	$P1 < 240$	$P2 \geq 80$	$P2 < 480$	$P1 \geq A1$ y $P1 < A2$	$P2 \geq Y3$ y $P2 < Y4$	Z	1110
Adelante	$P1 \geq 240$	$P1 < 300$	$P2 \geq 0$	$P2 < 50$	$P1 \geq A2$ y $P1 < A3$	$P2 \geq Y1$ y $P2 < Y2$	W	0001
Stop	$P1 \geq 240$	$P1 < 300$	$P2 \geq 50$	$P2 < 80$	$P1 \geq A2$ y $P1 < A3$	$P2 \geq Y2$ y $P2 < Y3$	Null	
Atrás	$P1 \geq 240$	$P1 < 300$	$P2 \geq 80$	$P2 < 480$	$P1 \geq A2$ y $P1 < A3$	$P2 \geq Y3$ y $P2 < Y4$	S	0010
Adelante derecha	$P1 \geq 300$	$P1 < 400$	$P2 \geq 0$	$P2 < 50$	$P1 \geq A3$ y $P1 < A4$	$P2 \geq Y1$ y $P2 < Y2$	E	1011
Derecha	$P1 \geq 300$	$P1 < 400$	$P2 \geq 50$	$P2 < 80$	$P1 \geq A3$ y $P1 < A4$	$P2 \geq Y2$ y $P2 < Y3$	D	0011
Atrás derecha	$P1 \geq 300$	$P1 < 400$	$P2 \geq 80$	$P2 < 480$	$P1 \geq A3$ y $P1 < A4$	$P2 \geq Y3$ y $P2 < Y4$	C	1100

6. ANÁLISIS DE RESULTADOS

Durante el proceso de integración se realizaron pruebas de la interfaz gestual y como se visualiza en la figura 5, en la parte inferior derecha la consola de comandos muestra la posición del objeto (atrás derecha), los valores de las coordenadas (X1, Y1, radio 1 y radio 2) y el punto central (P1

coordenada X centro y P2 coordenada Y centro). Estos cálculos obtenidos durante el tracking con la pelota azul se verifican en la tabla 2, los datos de P1 y P2 superan los valores límite a derecha y hacia abajo, por lo que claramente se encuentra en la posición "Atrás-Derecha". Este proceso es bastante fiable para determinar la posición del objeto seleccionado, ya que la precisión de los valores es aceptable para cuadrantes tan grandes y porque los valores críticos o de mayor interés son solo los centrales (intervalos de detección comando stop). El algoritmo diseñado permite identificar la orden que se envía al robot mediante la captura del centro del segmento de la imagen seleccionada, de acuerdo a las condiciones definidas para los intervalos de control de la interfaz gestual (ver tabla 2), el programa retorna una cadena de texto que indica la posición en la que se encuentra y con el comando stop

determina la posición del objeto seleccionado y envía la señal para que modifique la posición en el mundo virtual y en el servomecanismo (ver figura 5). El tiempo mínimo de respuesta de Python para comenzar a enviar los datos de control es de 1.5 segundos, debido a la instanciación de la clase serial. El objetivo es transferir por ondas de radio un dato de 4 bits hacia el circuito de control del servomecanismo, de manera que las órdenes que se imparten las realice el mismo. El área de trabajo no supera los 20 metros, es un enlace punto a punto, en la banda libre de radio frecuencia de 2.4 GHz y no requiere de repetidores. En el proceso de captura del dato, este debe mantenerse en el puerto al menos 500 ms, para que el microcontrolador interprete la señal como un pulso alto, que ocasiona un retraso mínimo entre el mundo virtual y el servomecanismo (ver figura 5).



FIGURA 5. Teleoperación del servomecanismo y del mundo virtual

Durante la fase de pruebas de la transmisión de datos del módulo de comunicación al servomecanismo, se detectaron problemas de baja potencia y se decidió eliminar los últimos cuatro datos de control que tuvieran la menor cantidad de bits en alto (ver tabla 1), como alternativa para controlar el consumo de potencia del sistema RF (Radio Frecuencia) en el circuito de control, se puede cambiar la banda de operación con soporte a arquitecturas abiertas que requieran de unidades de control multipunto para la teleoperación del servomecanismo y manejo del consumo de la batería cuando no se utiliza el robot. La fiabilidad de la posición de los objetos depende en gran

medida de la selección de los algoritmos, en este proyecto se utilizó CamShift y MeanShift para determinar el área que ocupa el objeto a partir de la retroproyección del histograma de la imagen y se calcula la probabilidad de la región (frame) semejante al objeto que sigue en movimiento. En el histograma de color se pueden verificar los valores obtenidos en el tracking y por los valores límites de la tabla 2 se valida la posición del objeto. En el prototipo desarrollado se utilizó un sistema de control a lazo abierto, que no ofreció la fiabilidad para que la señal de control actuara de acuerdo a lo requerido.

REFERENCIAS

1. Blender, «Supported Platforms,» 2011. [En línea]. Available: <http://www.blender.org/>.
2. G. R. Bradski, «Computer vision face tracking for use in a perceptual user interface,» 2010. [En línea]. Available: <http://download.intel.com/>.
3. R. Laganière, OpenCV 2 computer vision application programming cookbook over 50 recipes to master this library of programming functions for real-time computer vision, Packt Publishing, 2011.
4. Arduino, «Arduino,» 2011. [En línea]. Available: <http://www.arduino.cc/>.
5. G. Xin, H. Huan, J. Qing-Xuan, S. Han-Xu y S. Jing-Zhou, «3D augmented reality teleoperated robot system based on dual vision,» *Elsevier*, pp. 105-112, 2011.
6. G. Bradski y A. Kaebler, Learning openCV computer vision with the openCV library, O'Reilly Media, 2008.
7. J. Corrales, P. Gil, A. Candelas y F. Torres, «Diseño de una mini cámara motorizada para el seguimiento de objetos,» *Comité Español de Automática*, pp. 1-8, 2009.
8. C. Pradal, F. Boudon, C. Nouguié, J. Chopard y C. Godin, «PlantGL: A Python-based geometric library for 3D plant modelling at different scales,» *Elsevier*, pp. 1-21, 2009.
9. H. Thiruvengada, P. Derby, W. Foslien, J. Beane y A. Tharanathan, «The Influence of Virtual World Interactions toward Driving Real World Behaviors,» *Springer-Verlag Berlin Heidelberg*, p. 100–109, 2011.
10. L. Santamaría y J. F. Mendoza, «Construcción de mundos virtuales para el desarrollo de destrezas de lateralidad basado en Web3D,» *Educación en Ingeniería*, pp. 13-25, 2014.
11. M. Pesce, VRML para Internet, México: Prentice Hall, 1998.
12. J. Thompson, Videojuegos manual para diseñadores gráficos, Barcelona : Gustavo Gilli, 2009.
13. USTA Tunja, «Instrumentación y control,» 2011. [En línea]. Available: <http://www.ustatunja.edu.co/electronica/>.

7. CONCLUSIONES

Se desarrolló un algoritmo que recibe las señales de control de la interfaz gestual para ser transmitidas al microcontrolador del servomecanismo y realizar la teleoperación del modelo real y del avatar en el mundo virtual. En una segunda fase este algoritmo puede aportar en la implementación de un robot que pueda manipular objetos para la prevención de desastres naturales o facilitar el entrenamiento con el uso de simuladores. El funcionamiento del servomecanismo se puede optimizar a través de un sistema mecatrónico independiente para recibir la señal producto de la transmisión serial, que decodifique y convierta los datos en un arreglo de 4 bits, para controlar el movimiento de las ruedas y de la pinza del robot. La comunicación deberá ser full dúplex para hacer corrección y precisión en el control del servomecanismo.