UNIVERSIDAD NACIONAL DE INGENIERÍA
FACULTAD DE INGENIERÍA MECÁNICA
Escuela de Ingeniería Mecatrónica

DISEÑO DE UN SISTEMA DE NAVEGACIÓN ASISTIDA PARA PERSONAS CON DISCAPACIDAD VISUAL EMPLEANDO RECONSTRUCCIÓN 3D DEL AMBIENTE, APRENDIZAJE PROFUNDO Y SONORIZACIÓN

INFORME FINAL

Curso: Proyecto Mecatrónico
Alumnos: Centeno León, Santiago Cristhian
Campos Alarcón, Edwin Edmar
Profesor: M.Sc. Albites Sanabria, José Luis
Asesor: PhD. Cárdenas Lizana, Paul Antonio

FECHA DE ENTREGA

27 de setiembre de 2020

INFORME FINAL

I. SITUACIÓN PROBLEMÁTICA Y DEFINICIÓN DEL PROBLEMA

I.1. SITUACIÓN PROBLEMÁTICA

En junio del 2012 la OMS estimó que en el mundo existen aproximadamente 285 millones de personas con discapacidad visual, de las cuales 39 millones son ciegas (nula visión) y 246 millones presentan discapacidad visual moderada y grave (baja visión) [17]. Esta población enfrenta distintos desafíos y limitaciones en su actividad cotidiana, los cuales reducen considerablemente su independencia.

Ese mismo año se realizó en el Perú la Encuesta Nacional Especializada sobre Discapacidad, que luego fue publicada por el INEI en el año 2014, en dicha encuesta se registraron 801 mil personas con discapacidad visual de tipo permanente. Esto equivale al 2.6 % del total de la población peruana [18].

Según la encuesta [11], llevada a cabo en el Perú, los desafíos que los entrevistados desearían ver resueltos son las siguientes:

● Dificultad al cruzar pasos peatonales, en especial poder identificar el estado del semáforo.

● Dificultad al estimar la ubicación de escaleras y puertas.

● Obstáculos, baches en el camino.

● Dificultad para seguir la dirección del camino.

II. PROBLEMA GENERAL

En un país como el nuestro las limitaciones de movilidad que enfrenta la población con discapacidad visual en su actividad diaria son abundantes, especialmente en entornos desconocidos y dinámicos.

III. PROBLEMA INGENIERIL

¿Qué consideraciones de diseño mecatrónico y de diseño de software se deben adoptar con el fin de un dispositivo de asistencia que alivie parcialmente los desafíos de movilidad, orientación y reconocimiento del ambiente que enfrenta la población con discapacidad visual en el contexto de la ciudad de Lima? ¿Teniendo como restricciones un bajo precio y que la propuesta no dependa del acceso a internet?

IV. OBJETIVOS

IV.a. GENERAL

Diseñar un sistema de navegación asistida de bajo costo dirigido a personas con discapacidad visual que mejore su movilidad y orientación en contextos dinámicos, logrando reconstruir el entorno tridimensional y comunicarlo al usuario mediante el uso de procesamiento de imágenes, aprendizaje profundo, y sonorización del entorno. El dispositivo se diseñará para la población con deficiencia visual en Lima.

IV.b. OBJETIVOS ESPECÍFICOS

- Lograr la identificación de obstáculos en el rango de 0 a 5 metros, localizando los elementos espacialmente.
- Conseguir que el dispositivo identifique los siguientes elementos relevantes del ambiente: semáforos, escaleras, personas, pasos de cebra, con una precisión del 85 %.
- Lograr comunicar estos elementos del entorno al usuario mediante sonorización, incluyendo su localización y tamaño aproximado. El usuario debe reconocer los elementos con una precisión del 80 %.

V. DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

V.a. DESCRIPCIÓN

V.a.1. Diagrama de bloques pictórico

En el siguiente diagrama, los bloques rosas corresponden al subsistema de reconstrucción del entorno y los bloques púrpura al subsistema de interfaz humano-máquina.

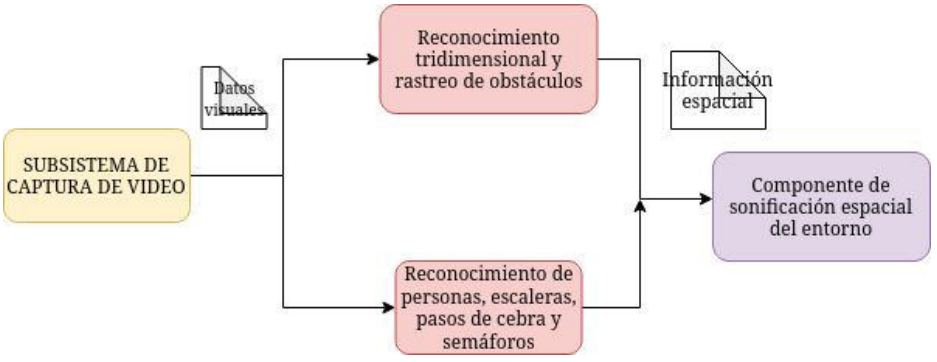


Figura 1. Diagrama de bloques del dispositivo de asistencia propuesto

V.a.2. Funcionamiento

El dispositivo de asistencia que se quiere diseñar está dirigido a mejorar la movilidad y orientación de personas con discapacidad visual en entornos cambiantes. La operación del sistema consiste en la adquisición de video, y su procesamiento en tiempo real para generar una representación del entorno, la que será percibida mediante sonorización espacial por el usuario (figura 2). Este funcionamiento se aprecia en la figura 1. La representación del entorno consiste en un componente de reconstrucción tridimensional e identificación de obstáculos; así como en componentes capaces de reconocer personas, escaleras, semáforos y pasos peatonales (figura 3).

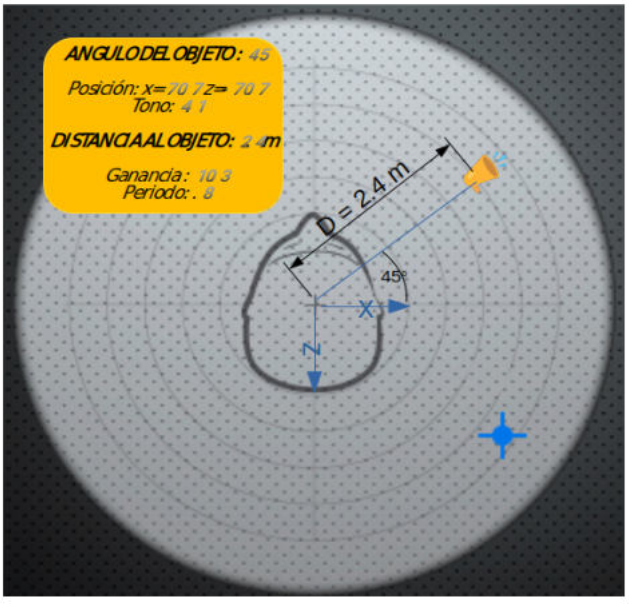


FIgura 2. Representación sonora de un elemento

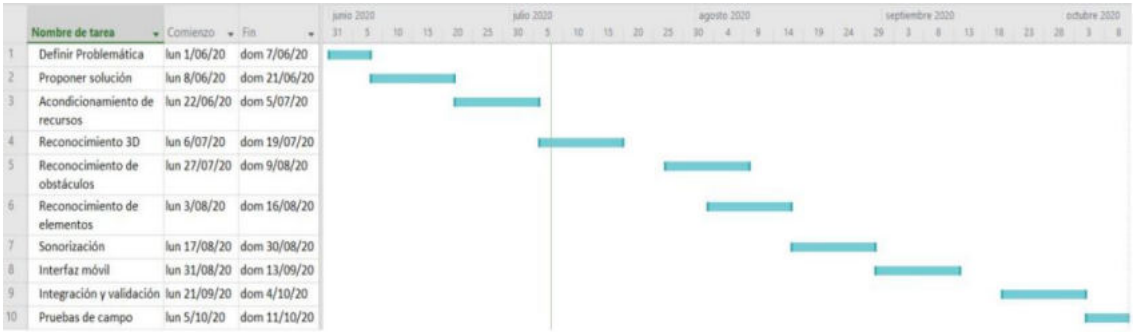
Figura 3. Detección de elementos relevantes

VI. CRONOGRAMA

Tabla 1

Actividad	*Aspectos a considerar*	Fechas
Definición de la problemática	Definición de la problemática, entrevistas con el público objetivo, y determinación de requerimientos de diseño	01/06/20 -07/06/20
Proponer una solución	Definir los componentes principales, así como recursos de hardware y software necesarios	08/06/20 – 21/06/20
Acondicionamiento de recursos técnicos	Instalar programas necesarios, librerías de procesamiento de imágenes.	22/06/20 – 05/07/20

Diseñar reconocimiento espacial 3D y reconocimiento de obstáculos	Rastrear objetos tridimensionales; estimando su ubicación y tamaño.	06/07/20 – 09/08/20
Componentes de reconocimiento de elementos relevantes	Reconocimiento de personas, semáforos, escaleras.	03/08/20 – 10/08/20
	Identificar la orientación de pasos peatonales.	11/08/20 – 16/08/20
Componente de sonorización	Representación de entorno mediante sonidos tridimensionales.	17/08/20 – 30/08/20
Integración y validación	Integración de las distintas funciones en la solución.	21/09/20 – 04/10/20
Pruebas de campo	Medición de desempeño por parte de usuarios	05/10/20 – 11/10/20
Presentación final	Informe final y presentación oral	



	Nombre de tarea	Comienzo	Fin
1	Definir Problemática	lun 1/06/20	dom 7/06/20
2	Proponer solución	lun 8/06/20	dom 21/06/20
3	Acondicionamiento de recursos	lun 22/06/20	dom 5/07/20
4	Reconocimiento 3D	lun 6/07/20	dom 19/07/20
5	Reconocimiento de obstáculos	lun 27/07/20	dom 9/08/20
6	Reconocimiento de elementos	lun 3/08/20	dom 16/08/20
7	Sonorización	lun 17/08/20	dom 30/08/20
8	Interfaz móvil	lun 31/08/20	dom 13/09/20
9	Integración y validación	lun 21/09/20	dom 4/10/20
10	Pruebas de campo	lun 5/10/20	dom 11/10/20

Figura 4. Diagrama de Gantt del proyecto

VII. AVANCES

RECONSTRUCCIÓN TRIDIMENSIONAL

a. OBJETIVOS DEL AVANCE

- Diseñar e implementar el componente de reconstrucción tridimensional del entorno.

- Calibrar el arreglo de cámaras estéreo; calculando los parámetros extrínsecos e intrínsecos.

b. SUSTENTO TEÓRICO

Visión estereoscópica

Es una tarea fundamental del área de visión artificial. Consiste en reconstruir una escena 3D a partir de las 2 vistas de una cámara estéreo. Para ello, primero se identifican puntos correspondientes en la imagen derecha e izquierda (*stereo matching)*; es decir, se hallan las proyecciones en ambas cámaras del mismo punto del escenario tridimensional [1].

Tras efectuar este proceso se obtiene un *disparity map*; donde para cada pixel, el desplazamiento entre la imagen izquierda y derecha es calculado. A partir de esta representación y algunos datos de calibración, se determina el mapa de profundidad.

Rectificación y corrección de distorsión (calibración)

Como se observa en la figura A.1, para cada pixel en la imagen base, su punto correspondiente se debe encontrar en la *línea epipolar.* El proceso de rectificación consiste en transformar las imágenes de manera que las líneas epipolares sean colineales y horizontales, a la vez que los ejes focales se vuelven perpendiculares a la línea que une los focos (*baseline)* como se ve en la figura A.2.

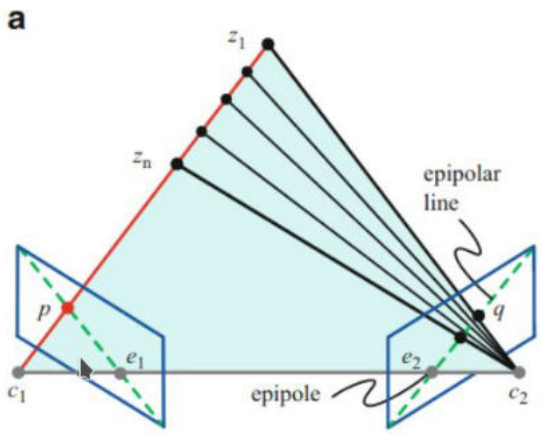


Figura A.1. Vistas previamente a la rectificación [1]

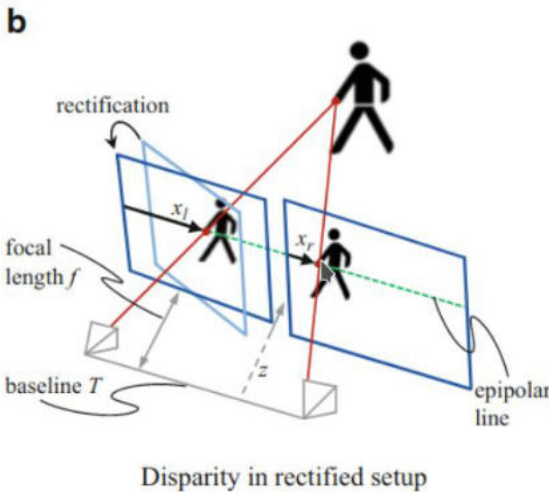


Figura A.2. Vistas tras la rectificación [1]

Además de rectificar las vistas, es necesario corregir las distorsiones generadas por defectos de las cámaras (distorsión radial y distorsión tangencial, figura A.3). Esto se puede efectuar mediante un mapeo inverso a partir de un método de interpolación.

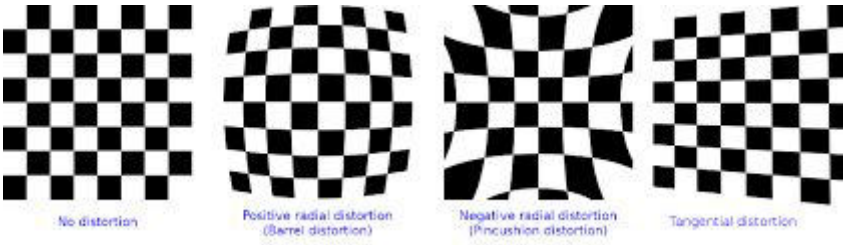


Figura A.3. Distorsiones causadas por defectos de cámara

Tras la rectificación y la corrección de distorsión, el mapa de profundidad se calcula mediante la siguiente expresión:

$$z = \frac{f\,T}{(x_l - x_r)}$$

Donde *z [m]* representa la distancia del punto tridimensional al centro del arreglo estéreo, *f [píxeles]* es la distancia focal, *T [m]* es la distancia entre focos, y *d [pixeles]* es la disparidad entre píxeles correspondientes.

Los procesos de rectificación y corrección de distorsión se pueden unir en una sola etapa de mapeo inverso.

Stereo matching

Los algoritmos modernos de *stereo matching* son métodos densos, que tratan de determinar la disparidad entre cualquier par de puntos en ambas vistas. Se dividen a grandes rasgos en métodos locales y globales [1].

Para el *stereo matching* es necesario en primer lugar el calcular el *matching cost C(x,y,d)* , que corresponde a la medida de similitud entre el pixel *(x,y)* en la imagen izquierda, y el *pixel (x-d,y)* en la derecha. Entre las métricas se encuentran las siguientes: *sum of squared intensity differences* (SSD), *sum of absolute differences* (SAD), *normalized cross correlation* (NCC).

Tras ello, los métodos locales proceden a la agregación de costos, basándose en restricciones como la permanencia de la disparidad o la segmentación por colores; estos métodos tienen una implementación eficiente. Los métodos globales, se enfocan en optimizar un cálculo de entropía que tiene como restricción la suavidad de la disparidad. Dichos algoritmos son más robustos y precisos, pero tienen un costo computacional elevado. Métodos locales con estrategias de agregación sofisticadas, o el algoritmo *Semi Global Optimization Strategy* logran un compromiso entre

eficiencia y desempeño. Este último optimiza la disparidad en múltiples direcciones para cada pixel.

Finalmente, se puede refinar el mapa de disparidad, interpolando para áreas inválidas y para espacios entre píxeles, eliminando valores atípicos, filtrando el mapa, etc.

Semi Global Optimization Algorithm

Estrategia que combina una *rank transform* para una determinación inicial de los *matching costs,* y un método global para reducir cálculos erróneos y asegurar una mayor consistencia en los resultados.

Los costos iniciales se determinan según la siguiente expresión:

$$C(p_x, p_y, d) = |R_b(p_x, p_y) - R_m(p_x - d, p_y)|$$

Donde el costo para un punto *p* y una disparidad *d* se halla como la diferencia absoluta entre la transformada *rank* (*R*) para el punto *p* en la vista izquierda y la transformada para el punto correspondiente (desplazado *d* unidades) en la vista derecha.

La transformada *rank* se define como el número de píxeles en una vecindad cuadrangular de MxM unidades (*A(p)*) cuya luminosidad es menor que la luminosidad del punto *p* (*I(p)*).

$$R(p_x, p_y) = ||p' \in A(p),\ tal\ que\ I(p') < I(p)||$$

Con el fin de mejorar la estimación inicial, se introducen restricciones de consistencia globales. Esto se consigue agregando costos a lo largo de varias direcciones *r.* La disparidad con la menor suma de costos agregados se selecciona como resultado del proceso de *stereo matching* para el punto *(px,py)*.

Costo en una ruta *r:*

$$R(p_x, p_y) = ||p' \in A(p), \ tal\ que\ I(p') < I(p)||$$

Suma de costos agregados:

$$S(p_x, p_y, d) = \sum_r L_r(p_x, p_y, d)$$

Selección de disparidad final:

$$d^* = min_d\, S(p_x, p_y, d)$$

c. ANÁLISIS

Para la calibración de las cámaras; se hace uso del proceso mencionado en [2]. El cual consiste en tomar capturas de un tablero de ajedrez, reconocer las intersecciones sobre cada imagen (Figura A.4); y al conocer las coordenadas 3D, se procede a calcular los parámetros intrínsecos de calibración.

Figura A.4. Calibración de cámaras

En el caso de la calibración estéreo, se lleva a cabo un proceso similar: se reconocen las esquinas en la vista izquierda y derecha; y a partir de las coordenadas del tablero en 3 dimensiones se estiman los parámetros de rectificación estéreo (Fig. A.5).

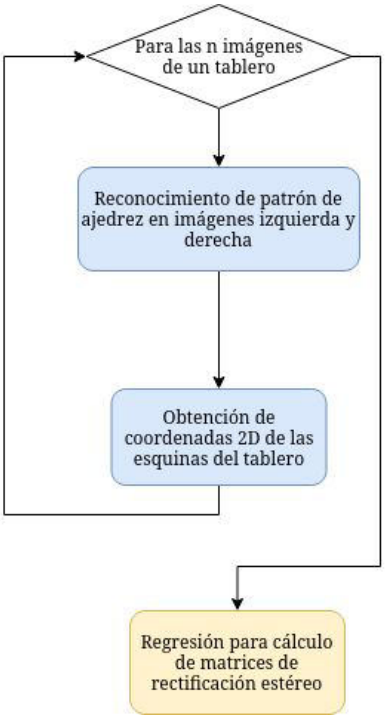


Figura A.5. Calibración estéreo

De esta forma, para acondicionar el par de capturas provenientes de las cámaras basta con introducir un mapeo que combine la corrección de distorsiones y la rectificación estéreo.

Como algoritmo de *stereo matching*, se adopta el método de *Semi Global Matching (SGM),* debido al balance que presenta entre eficiencia computacional y desempeño.

A continuación se presenta el flujo del algoritmo SGM, incluyendo la etapa de estimación inicial de costos mediante una transformada *rank,* y la etapa de agregación de costos mediante *semi global matching* propiamente dicho (Fig. A.6).

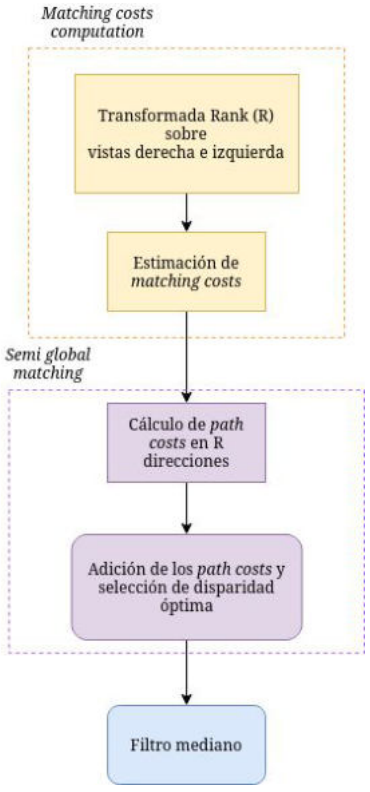


Figura A.6. Proceso de estimación de disparidad

Tras obtenerse el mapa de disparidad de la escena, se divide el campo de visión en 6 franjas horizontales, y a cada una se le asigna un valor de disparidad de acuerdo a la región más cercana. Dichos valores se convierten en distancias, las que se pueden representar en la imagen acústica para que el usuario las perciba. Esto se muestra en la Figura A.7.

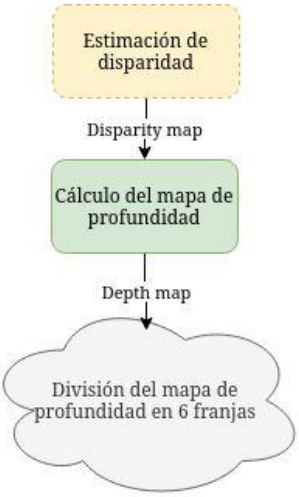


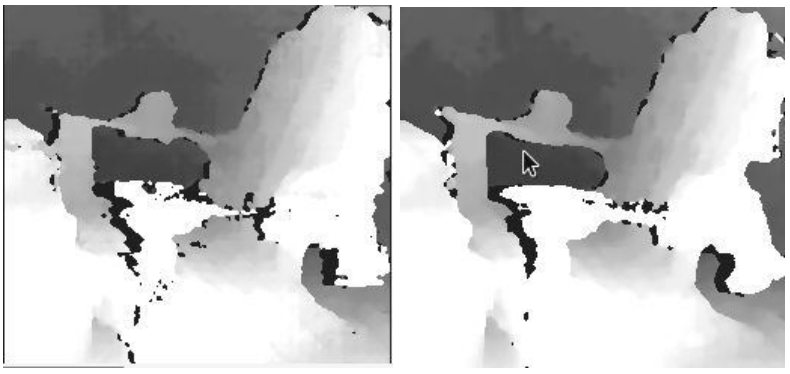
Figura A.7. Obtención de mapa de obstáculos a ser presentado al usuario

d. RESULTADOS, MEDICIONES Y VALIDACIÓN

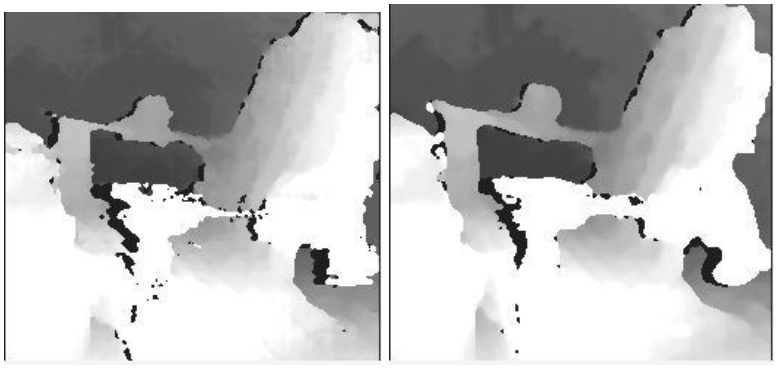
El algoritmo de reconstrucción es implementado con la librería *openCV.* El tiempo aproximado de ejecución del algoritmo en el hardware es de .02 segundos para una imágenes de 320x240 píxeles.

Como prueba de funcionalidad, para las vistas mostradas en la Fig. A.8 se calculan mapas de disparidad con distintos parámetros de reconstrucción (Fig. A.9) .

Figura A.9. Vistas izquierda y derecha de un escenario



i *ii*



iii *iv*

Figura A.10. Mapas de disparidad con distintos parámetros. *i: block Size* 3, *uniqueness ratio* 5. *ii:* 5,5. *iii:* 3,3. *iv:* 7,3.

e. PROBLEMAS PRESENTADOS

Para mejores resultados en la estimación de disparidad, es preferible utilizar cámaras idénticas alineadas horizontalmente. Fue necesario comprar 2 cámaras del mismo modelo.

f. CONCLUSIONES

Fue posible la implementación de un algoritmo de reconstrucción estéreo a un bajo coste computacional.

El proceso de calibración permitió rectificar las imágenes derecha e izquierda, alineando horizontalmente los puntos correspondientes. Este fue un paso necesario para asegurar la calidad de la posterior reconstrucción.

g. BIBLIOGRAFÍA

1. BANZ C. [et al.] *Architectures for Stereo Vision*. 2019, Handbook of Signal Processing Systems, Springer.

2. *Camera calibration.* Open CV documentation.

Disponible desde:

https://docs.opencv.org/3.4/dc/dbb/tutorial_py_calibration.html

INTERFAZ AUDITIVA

a. OBJETIVOS DEL AVANCE

- Diseñar e implementar el componente de interfaz auditiva entre el usuario y el dispositivo.

- Diseñar el algoritmo que represente la información espacial mediante sustitución sensorial auditiva.

b. SUSTENTO TEÓRICO

Sonificación

Se refiere a representar información mediante audio no verbal. En el ámbito de los sistemas de asistencia para invidentes, es empleada para funciones de evasión de obstáculos y navegación [1].

Substitución sensorial

Consiste en expandir la percepción de una persona, de manera que esta pueda sensar información ambiental correspondiente a un sentido (generalmente la vista) mediante otros (generalmente el tacto y la audición). En este caso, se transmite información espacial a través de representaciones sonoras [2].

HRTF (*Head related transfer function)*

Consiste en la relación entre la transformada de Fourier de la señal sonora como es percibida en el tímpano del oyente y la transformada de la señal

que llegaría al centro de la cabeza si el oyente estuviera ausente. Se emplea para sintetizar escenas auditivas que dan la impresión de suceder alrededor del oyente [3].

Principales estrategias de sonorización

En [1], se reseñan las principales estrategias de representación sonora en aplicaciones de asistencia para invidentes:

1. El dispositivo vOICe consiste en una transformada de Fourier inversa; donde cada columna de pixeles corresponde a un instante de la señal y la posición vertical de cada pixel corresponde a una frecuencia. Se orienta a representar imágenes 2D, mas se podría adaptar a aplicaciones de representación 3D.

2. El sistema EAV genera una mapa de profundidad a baja resolución (16x16x16) en el que cada voxel (cubo) se convierte en una fuente sonora ubicada espacialmente; creando la ilusión de una nube de sonidos.

3. El sistema *Cognitive Aid System for Blind System* representa un mapa de profundidad 1D es mediante un sonido que se mueve horizontalmente; donde la distancia corresponde al volumen y la dirección se transforma en la posición estéreo de la fuente de sonido.

4. *EyeMusic* representa una imagen a color como un espectrograma musical; donde cada columna corresponde a un momento, cada coordenada vertical corresponde a una nota; y cada color se convierte en un instrumento diferente.

5. El dispositivo Naviton hace un paneo frontal de la escena y representa los objetos y las paredes como fuentes sonoras HRTF, donde la distancia se codifica a volumen y tono; el tamaño de los objetos corresponde a la duración del sonido; además, para a categoría de objeto le corresponde un timbre.

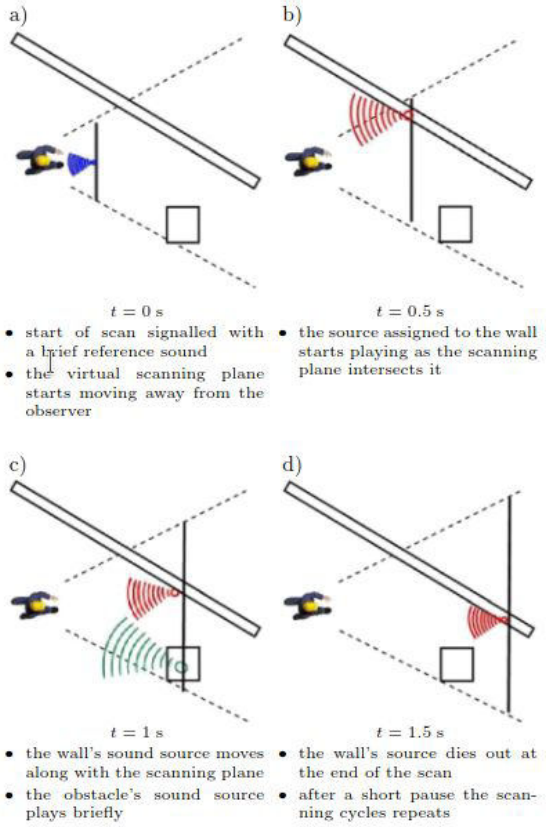


Figura B.1. Representación sonora efectuada por Naviton [1]

c. ANÁLISIS

Algunas estrategias de representación auditiva convierten un mapa de profundidad o una imagen 2D en un espectrograma donde la coordenada vertical corresponde a una frecuencia o tono. Otros métodos consisten en

transformar un mapa de profundidad unidimensional en un sonido cambiante donde la distancia corresponde al volumen o timbre. Estrategias comunes representan mapas de profundidad mediante la síntesis HRTF de fuentes de sonido; codificando la distancia y otra información de los objetos en las características acústicas.

Para el presente dispositivo, se decidió adoptar esta tercera estrategia; debido a que permite no saturar la capacidad perceptual del usuario, y representar solo los elementos más relevantes en su entorno inmediato.

En específico, para la detección de obstáculos, se hará un barrido horizontal del entorno como se muestra en la figura B.2, dividiendo el panorama en 6 segmentos. En cada segmento se reproducirá un timbre ubicado tridimensionalmente, cuyo volumen y duración representa la distancia al obstáculo. Mientras más cerca de un obstáculo se encuentre el usuario, el barrido será más frecuente. La representación acústica detallada es expresada en las relaciones (1).

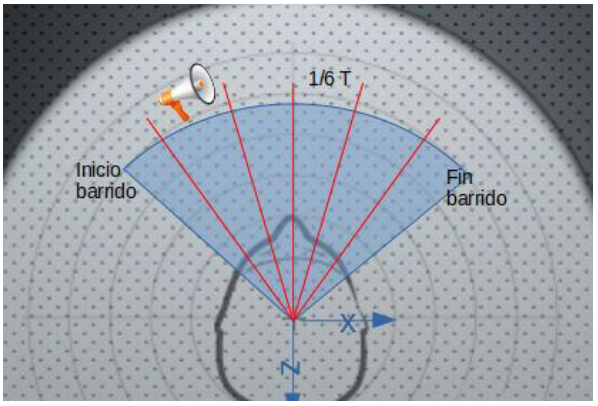


Figura B.2. Barrido horizontal del entorno.

$$x = 50\,cos(\alpha) \quad y = 0 \quad z = -50\,sin(\alpha) \quad x,y,z: \text{ posición señal}, \ \alpha: \text{ ángulo de objeto}$$

$$gain = 108 - 77.2\,d + 19.3\,d^2 - 1.62\,(d^3) \quad gain: \text{ ganancia}, \ d: \text{distancia a objeto}$$

$$length = \{2.5, \text{ si distancia en } < 0.39;\ 1m > ...\} \quad length: \text{ duración de señal}$$

$$T = \{0.78, \text{ si distancia en } < 0.39;\ 1m > ...\} \quad T: \text{ periodo}$$

Relaciones 1. Representación acústica de elementos detectados

Para la detección de objetos, cada elemento se representa como un sonido 3D periódico cuyo timbre corresponderá a una categoría de objeto. En esta, la distancia se codifica como volumen y periodo; mientras que la posición del elemento se evidencia en la ubicación espacial de la fuente acústica y el tono de la señal. Esto se expresa en las siguientes relaciones (2):

$$x = 50\,cos(\alpha) \quad y = 0 \quad z = -50\,sin(\alpha) \quad x,y,z: \text{ posición señal}, \ \alpha: \text{ ángulo de objeto}$$

$$gain = 108 - 77.2\,d + 19.3\,d^2 - 1.62\,(d^3) \quad gain: \text{ ganancia}, \ d: \text{distancia a objeto}$$

$$pitch = 1.5\,\alpha + 0.5 \quad pitch: \text{ tono de señal}$$

$$T = \{0.4, \text{ si distancia en } < 0.39;\ 1m > ...\} \quad T: \text{ periodo}$$

Relaciones 2. Expresión de representación acústica de obstáculos

El posicionamiento tridimensional de fuentes acústicas se lleva a cabo mediante síntesis HRTF.

El programa encargado de la representación auditiva corre en segundo plano y funciona de servidor para el programa principal del dispositivo. La implementación de las señales 3D se logra con el módulo *openAL*.

d. RESULTADOS, MEDICIONES Y VALIDACIÓN

Se les hizo escuchar la representación acústica de un entorno simulado a varios usuarios pidiéndoles que identifiquen las direcciones y distancias a los elementos. Al entrevistar a los usuarios, se comprobó la correcta discriminación de posición y distancia para la detección de obstáculos; así como la diferenciación entre la representación de obstáculos y la representación de elementos. Esto se muestra en la tabla B.1.

	Situación A: 1 obstáculo (posición 3)	Situación B: 2 obstáculos (posiciones 2,5)	Situación C: 2 obstáculos (posiciones 2,4)y 1 elemento (posicion 2)
Usuario 1	posición 3	posiciones 3,5	posiciones 2,5,1
Usuario 2	posición 2	posiciones 1,5	posiciones 2,5,2
Usuario 3	posición 3	posiciones 2,5	posiciones 2,4,1
Usuario 4	posición 3	posiciones 2,6	posiciones 3,5,3

Tabla B.1. Resultados a la prueba de reconocimiento del entorno

Un clip de audio que simula la interfaz con el usuario se encuentra en *test.mp4*.

e. PROBLEMAS PRESENTADOS

Al principio se pensó adoptar un método de representación ligeramente diferente, donde cada elemento y obstáculo se representaba como una

fuente sonora que se repetía cada cierto tiempo. Esto fue descartado por consideraciones de contaminación auditiva, poca eficiencia (se reproducen muchos sonidos para representar un objeto) y problemas de discriminación de señales.

f. CONCLUSIONES

Tras emprender un sondeo por los métodos de representación auditiva usados en dispositivos de asistencia; se adoptó la estrategia de representar los elementos detectados como fuentes sonoras sintetizadas con HRTF. La representación acústica de los obstáculos se logró efectuando un barrido horizontal; mientras que la representación de los elementos a detectar consistió en timbres periódicos.

A través de la prueba de reconocimiento del entorno virtual se comprobó la efectividad del sub sistema para interactuar con los usuarios y comunicarles información ambiental.

Al integrar el componente en el sistema principal, se debería volver a realizar las pruebas de reconocimiento del entorno; y efectuar ajustes de acuerdo a ellas.

g. BIBLIOGRAFÍA

1. BUJACZ, M.; STRUMILLO, P. *Sonification: Review of Auditory Display Solutions in Electronic Travel Aids for the Blind*. 2016 IPPT PAN.

2. ELLI G.V., BENETTI S., COLLINGTON O.(2014). *Is There a Future for Sensory Substitution Outside Academic Laboratories?* Multisensory Research.

3. ZOTKIN, D. [et al.]. *HRTF personalization using anthropometric measurements*. 2003 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics.

DETECCIÓN DE OBJETOS

a. OBJETIVOS DEL AVANCE

- Reconocer personas, semáforos, escaleras que son detectadas a través de las cámaras del dispositivo de asistencia.

- Identificar la presencia de pasos peatonales.

- Medir en lo posible el uso del costo computacional para poder volverlo un dispositivo portable y autónomo.

b. SUSTENTO TEÓRICO

Detección de objetos

Esta tecnología se relaciona mucho con la visión artificial y el procesamiento de imágenes.

Su principal uso es detectar casos de objetos semánticos de una cierta clase ya sea en videos o imágenes. Durante la detección se debe tener en cuenta

las características especiales propias de cada clase de objeto ya que esto será de ayuda en la clasificación y detección [1].

Sistema de tiempo real

Es un sistema que generalmente interacciona con el entorno físico y responde a los estímulos del entorno dentro de un plazo de tiempo determinado ya que una respuesta tardía a un suceso externo puede tener consecuencias fatales como por ejemplo un choque en un automóvil autónomo [2].

Redes Neuronales Convolucionales

Este tipo de red es una variación de un perceptrón multicapa, donde las aplicaciones generales son en matrices bidimensionales como imágenes digitales.

Dentro de su arquitectura encontramos múltiples capas de filtros convolucionales de una o más dimensión, aunque después de cada capa se suele añadir una función no lineal.

Son muy usadas por su buena efectividad para tareas de visión artificial , así como la clasificación y segmentación de imágenes [3]

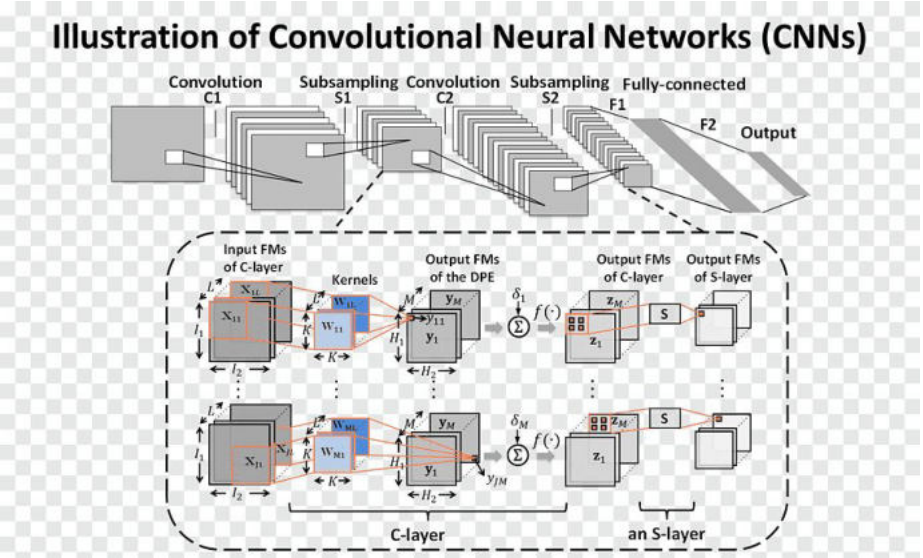
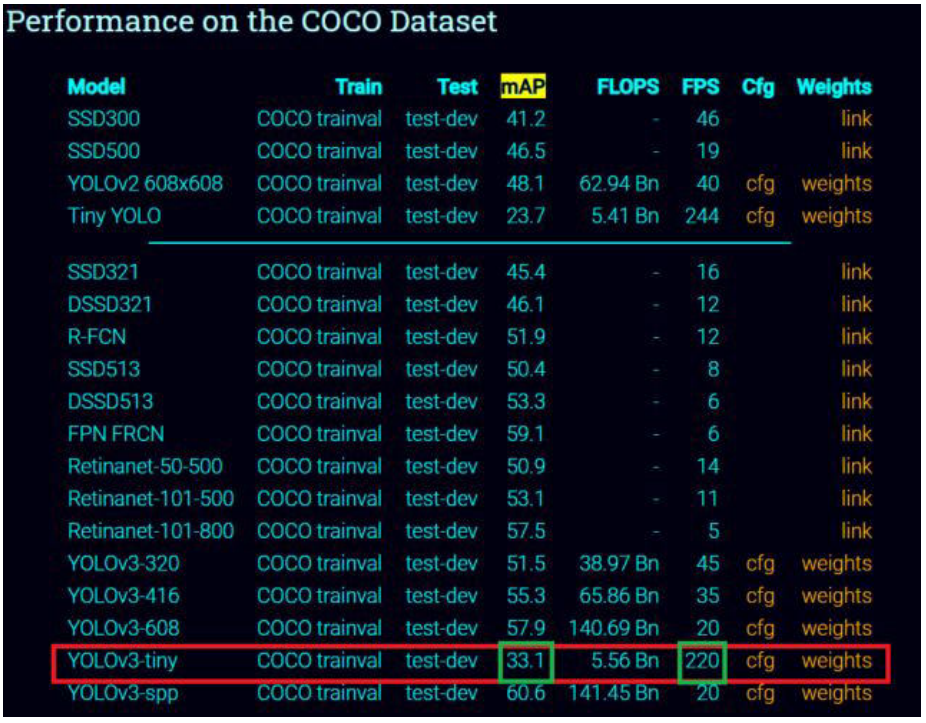


Figura C.1. Arquitectura de una red neuronal convolucional.

c. ANÁLISIS

Entre los modelos de detección en tiempo real disponibles se decidió adoptar el modelo tinyYOLOv3, por su alta precisión y su valor mAP, además de su alta eficiencia computacional puesto que su tiempo de inferencia es menor respecto a otros modelos de detectores y sus cuadros por segundo en detección de videos son elevados [4]; como se puede observar en las siguientes figuras comparativas.



Performance on the COCO Dataset

Model	Train	Test	mAP	FLOPS	FPS	Cfg	Weights
SSD300	COCO trainval	test-dev	41.2	-	46		link
SSD500	COCO trainval	test-dev	46.5	-	19		link
YOLOv2 608x608	COCO trainval	test-dev	48.1	62.94 Bn	40	cfg	weights
Tiny YOLO	COCO trainval	test-dev	23.7	5.41 Bn	244	cfg	weights
SSD321	COCO trainval	test-dev	45.4	-	16		link
DSSD321	COCO trainval	test-dev	46.1	-	12		link
R-FCN	COCO trainval	test-dev	51.9	-	12		link
SSD513	COCO trainval	test-dev	50.4	-	8		link
DSSD513	COCO trainval	test-dev	53.3	-	6		link
FPN FRCN	COCO trainval	test-dev	59.1	-	6		link
Retinanet-50-500	COCO trainval	test-dev	50.9	-	14		link
Retinanet-101-500	COCO trainval	test-dev	53.1	-	11		link
Retinanet-101-800	COCO trainval	test-dev	57.5	-	5		link
YOLOv3-320	COCO trainval	test-dev	51.5	38.97 Bn	45	cfg	weights
YOLOv3-416	COCO trainval	test-dev	55.3	65.86 Bn	35	cfg	weights
YOLOv3-608	COCO trainval	test-dev	57.9	140.69 Bn	20	cfg	weights
YOLOv3-tiny	COCO trainval	test-dev	33.1	5.56 Bn	220	cfg	weights
YOLOv3-spp	COCO trainval	test-dev	60.6	141.45 Bn	20	cfg	weights

Figura C.2. Comparación de modelos detectores en el Dataset COCO.

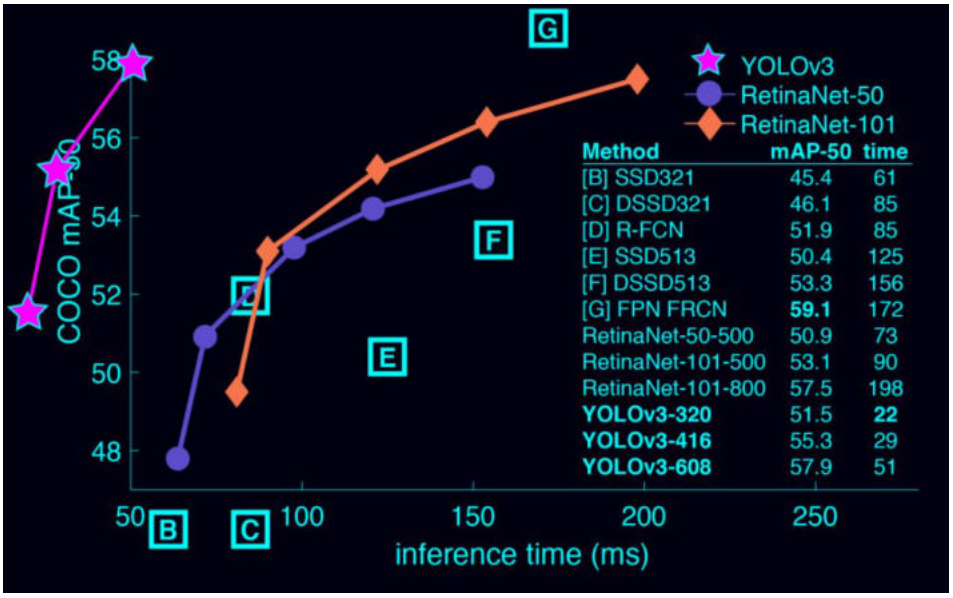


Figura C.3. Tiempo de inferencia comparativo de modelos de detectores.

Detección de personas y semáforos

Puesto que se usó el modelo tinyYOLOv3, este ya integra 88 clases pre entrenadas entre las cuales abarca la detección de personas y semáforos, pero no abarca la detección de pasos de cebra ni la detección de escaleras.

En la Figura C.4 y C.5 vemos el resultado de algunas de las clases detectadas que engloba el modelo mencionado.

Figura C.4. Detección de personas en una plaza de Lima

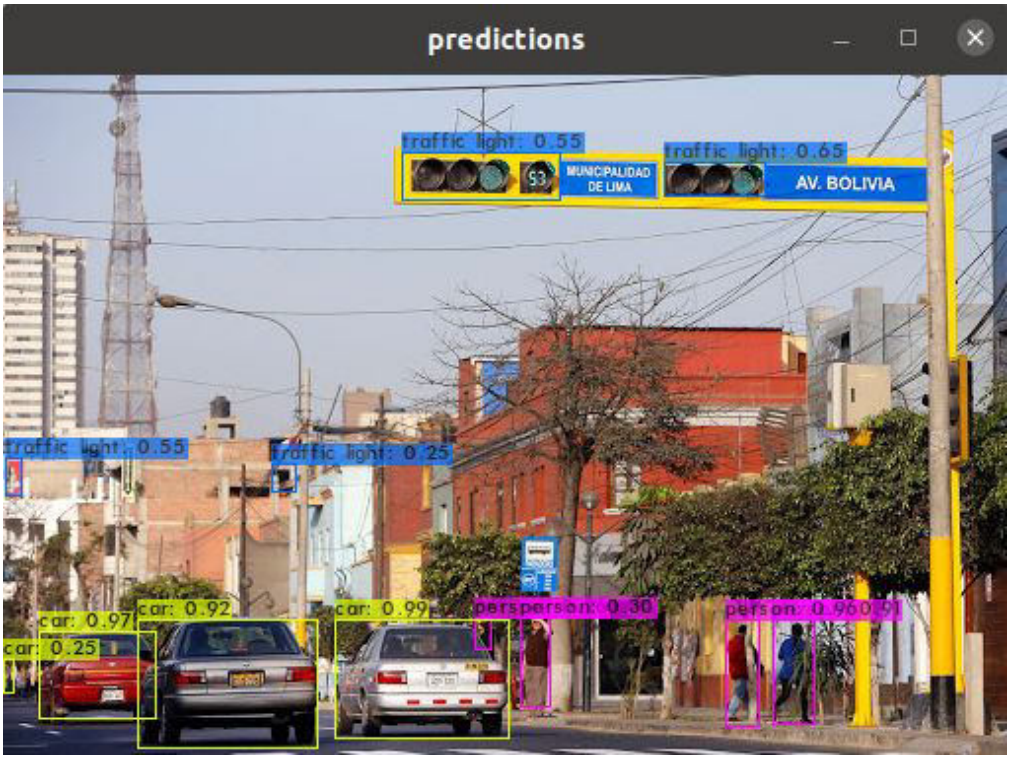
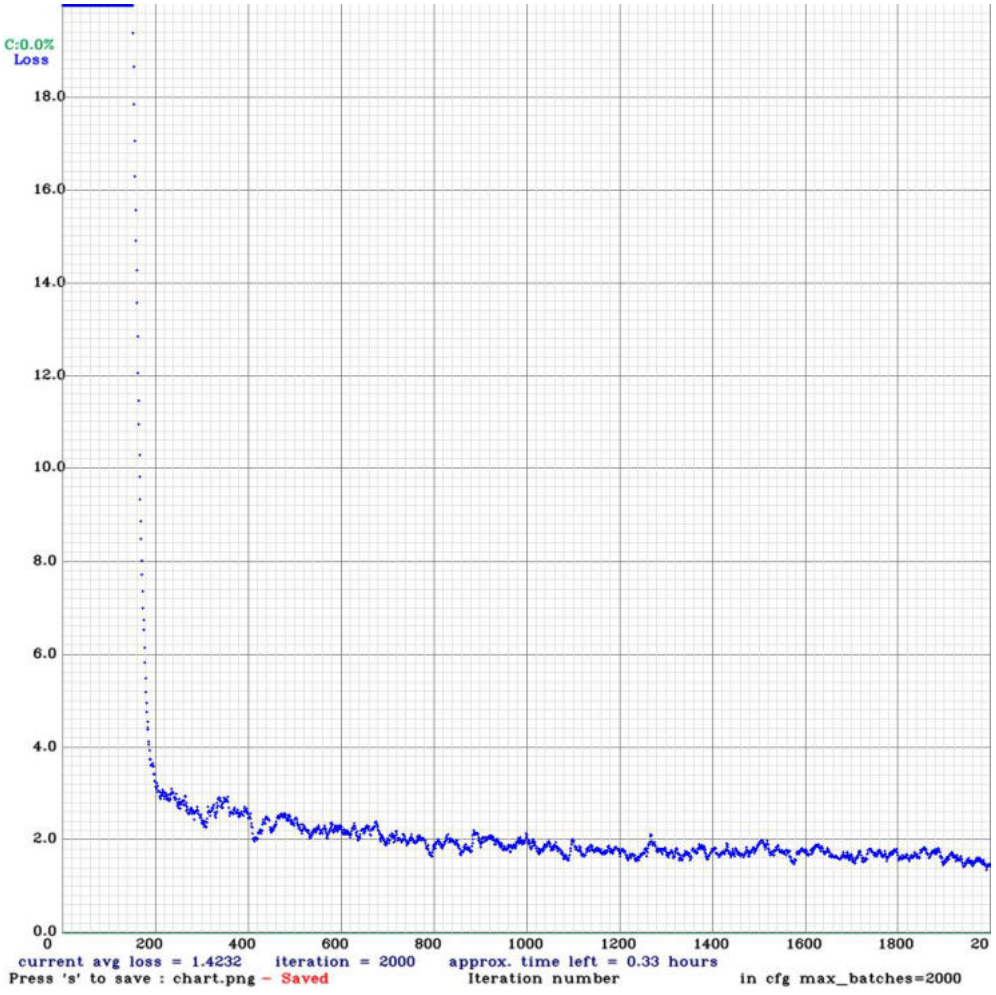


Figura C.5.Detección de semáforos en una avenida de Lima.

Detección de escaleras

Para la detección de escaleras se decidió entrenar el mismo modelo [5] con un dataset de 1488 imágenes [6], se decidió usar 2000 iteraciones para el entrenamiento (Ver Figura C.6.) puesto que entrenar con hacer más iteraciones corríamos el riesgo de caer en el overfitting. Una vez cargado los pesos obtenidos a la carpeta Darknet se obtuvieron como resultado del modelo entrenado para escaleras las figuras C.7 y las figuras C.8



nte:

Figura C.6. Modelo entrenado para escaleras luego de 2000 iteraciones.

Figura C.7. Detección de escaleras en una ciudad joven

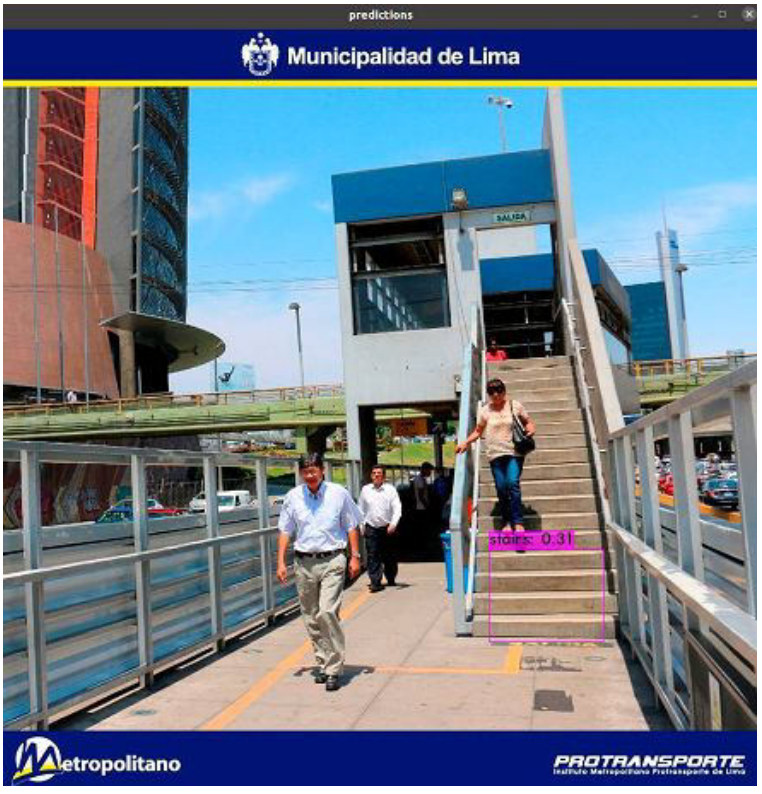


Figura C.8. Detección de escaleras en el metropolitano

Predicción del estado del semáforo detectado.

Para la predicción del color de semáforo, se decidió segmentar por colores el recuadro de detección, y de acuerdo al color que esté más presente ya sea (verde o rojo) se predice un estado. El rango considerado para el color rojo fue entre los valores *BGR* (0,0,170) y (80,80,255). En el caso del color verde, entre los valores (0,170,0) y (80,255,80). Esto se observa en la Fig. C.9.

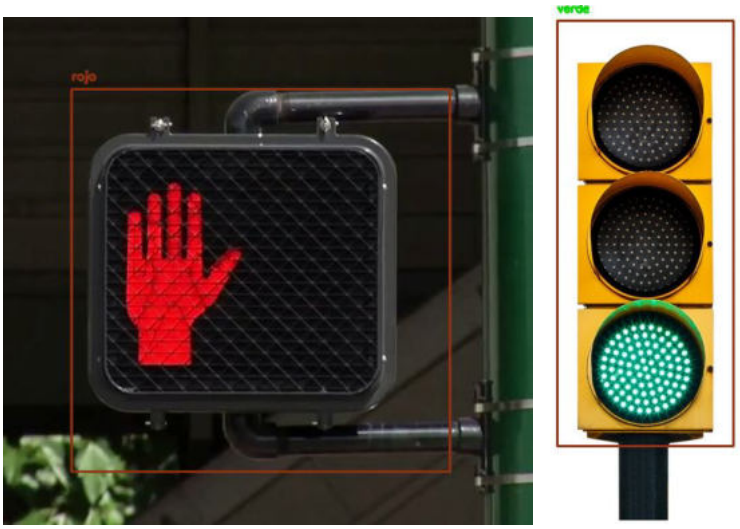


Figura C.9. Clasificación de colores de semáforos

Detección de pasos peatonales

Para la detección de pasos peatonales se decidió detectar la presencia de pasos peatonales de manera binaria (presente/ausente). La data consistió en imágenes de pasos de cebra en distintas posiciones; así como ejemplos negativos (Fig. C.10). Dichas muestras se extrajeron de [7].

Figura C.10. Data referida a pasos peatonales

En lo referente a la arquitectura del modelo, se decidió redimensionar las muestras a un formato (40,60,1); y aplicar sobre las imágenes 2 capas convolucionales, cada una seguida por una capa de *pooling* y una de regularización (*dropout*). Finalmente, se tienen 2 capas completamente conectadas (*fully connected layers)*. La arquitectura completa se observa en la Fig. C.11.

Layer (type)	Output Shape	Param #
conv2d_17 (Conv2D)	(None, 40, 60, 64)	1664
max_pooling2d_17 (MaxPooling	(None, 20, 30, 64)	0
dropout_24 (Dropout)	(None, 20, 30, 64)	0
conv2d_18 (Conv2D)	(None, 14, 24, 32)	100384
max_pooling2d_18 (MaxPooling	(None, 7, 12, 32)	0
dropout_25 (Dropout)	(None, 7, 12, 32)	0
flatten_20 (Flatten)	(None, 2688)	0
dense_45 (Dense)	(None, 50)	134450
dropout_26 (Dropout)	(None, 50)	0
dense_46 (Dense)	(None, 2)	102

Figura C.11. Arquitectura del modelo convolucional para detección de pasos peatonales

Tras efectuar 16 *epochs* de entrenamiento; la eficacia del modelo alcanzó una precisión del 91.81 % sobre el set de validación.

Detección de objetos en tiempo real

El programa principal captura video en tiempo real y aplica sobre los *frames* inferencias con los modelos de detección. Se emplea la librería *openCV* para la implementación de los modelos *tinyYOLOv3* (detección de personas, semáforos, y escaleras)*;* y la librería *tensor flow* para la implementación del modelo clasificador de pasos de cebra.

Integración de todas las funciones

El programa principal integra la función de reconstrucción 3D; así como la función de detección de elementos, y presenta la información recogida del entorno al usuario a través de la interfaz auditiva. Cada cierto número de frames, efectúan inferencias tanto con el modelo *tiny YOLO v3,* como con el modelo clasificador de pasos de cebra; así mismo, regularmente se actualiza la reconstrucción tridimensional del ambiente.

De acuerdo a la información espacial se reproduce la imagen sonora que sirve de representación. El campo de visión se divide en 6 para indicar la presencia de obstáculos mediante timbres; por otro lado notas musicales ubicadas estereoscópicamente indican la ubicación y naturaleza de los elementos relevantes: personas, escaleras, semáforos o pasos peatonales.

La estructura del programa principal se describe como un diagrama de flujo en la Fig. C.12.

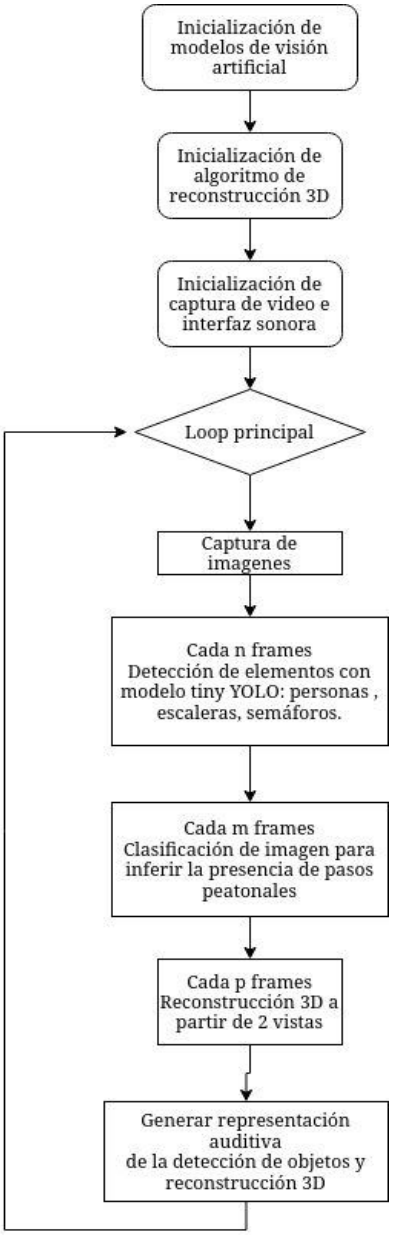


Figura C.12. Diagrama de flujo integrando todas las funciones

d. RESULTADOS, MEDICIONES Y VALIDACIÓN

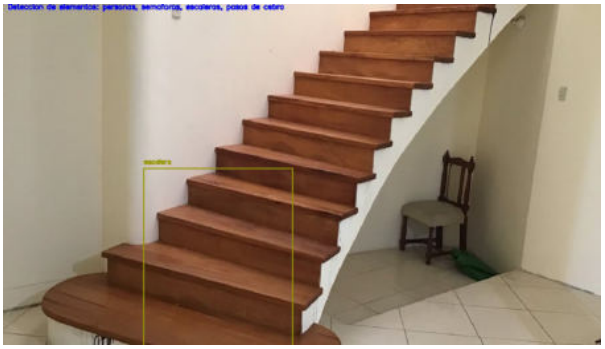
Las distintas funciones de detección se pusieron a prueba con varias imágenes tomadas en la ciudad de Lima. La precisión del sistema para detectar los distintos elementos se muestra en la Tabla C.1.

Objeto a Detectar	Imágenes Test	Eficiencia
Personas	10	100%
Semáforo	10	90%
Escaleras	10	80%
Pasos de cebra	Set de Validación	91.2%

Tabla C.1. Desempeño de las distintas funciones.

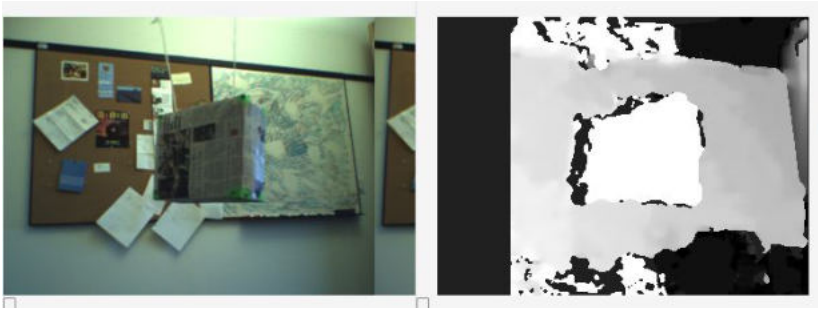
Tras llevarse a cabo la integración de las funciones en un solo sistema, la Figura C.13. muestra algunos cuadros de información espacial.

i



ii

iii



iv

Figura C.13. Funcionalidades del sistema de asistencia. *i:* reconocimiento de personas y semáforos. *ii:* reconocimiento de escaleras. *iii:* reconocimiento de pasos peatonales. *iv:* reconstrucción 3D.

e. PROBLEMAS PRESENTADOS

Los *dataset* encontrados de escaleras en internet tienen la peculiaridad de ser escaleras de plazas de europa lo que hace que nuestro dataset demuestre peor desempeño en el contexto local, debido a esto se decidió tener un bajo umbral *threshold* para la detección de esta clase.

En el reconocimiento de personas podemos mencionar dos problemas, el primero es cuando en el entorno hay una multitud, generando que en algunos casos no se llegue a detectar algunas de las personas o incluso se detecte más personas donde hay sólo una, el segundo es cuando las personas están muy alejadas el desempeño en la detección es pésimo, esto limitaría una detección de personas a larga distancia.

f. CONCLUSIONES

Fue posible el entrenamiento usando como base el modelo CNN *Tiny Yolo v3* para la detección de escaleras, personas, semáforos siendo este modelo de baja carga computacional como se propuso en los objetivos de este avance.

Por otro lado, para el reconocimiento de pasos peatonales se entrenó un modelo clasificador CNN con una precisión del 91.2 %.

Se llevó a cabo la integración de funciones en un programa principal; el que se pudo implementar y poner a prueba su funcionalidad.

Pruebas de uso que registren el desempeño y aprendizaje por parte de los usuarios constituyen una perspectiva de mejora para futuros trabajos.

g. BIBLIOGRAFÍA

1. Wikipedia: Object Detection [Ingreso 27/09/2020]

Disponible en: https://en.wikipedia.org/wiki/Object_detection

2. Wikipedia: Sistema de tiempo real [Ingreso 27/09/2020]

Disponible en: https://es.wikipedia.org/wiki/Sistema_de_tiempo_real

3. Wikipedia: Redes neuronales convolucionales [Ingreso 27/09/2020]

Disponible en:

https://es.wikipedia.org/wiki/Redes_neuronales_convolucionales

4. *YOLO: Real-TIme Object Detection.* [consultado el 27/09/2020]. Disponible en: https://pjreddie.com/darknet/yolo/

5. How to train tiny-yolo (to detect your custom objects). [consultado el 27/09/2020].

https://github.com/AlexeyAB/darknet#how-to-mark-bounded-boxes-of-objects-and-create-annotation-files

6. *Real Time stair detection-dataset* [Ingreso 27/09/2020] . Disponible desde:

https://akshayk07.weebly.com/real-time-stair-detection.html

7. SILVA, D. *Crosswalk-dataset* [Ingreso 07/09/2020]. Disponible desde:

https://www.kaggle.com/davidsilvam/crosswalkdataset/version/2

VIII. PRESUPUESTO ECONÓMICO DEL PROYECTO

Los costos se dividen en costos materiales, componentes y mano de obra:

COSTOS MATERIALES Y COMPONENTES

Tabla 2

Elemento	Costo unitario	Cantidad	Costo total
Casco	S./ 150	1	S./ 150
Cámara web	S./ 50	2	S./ 100
Audífonos inalámbricos	S./ 50	1	S./ 50
Computador portátil	S./ 1000	1	S./ 1000
Total			S./ 1300

MANO DE OBRA

Tabla 3

Concepto	Horas
Diseño e implementación	200

IX. CONCLUSIONES Y COMENTARIOS FINALES

Se presentó el planteamiento, objetivos y plan de actividades correspondientes al proyecto de investigación.

Se desarrollaron e implementaron las funcionalidades de reconstrucción tridimensional, detección de objetos y representación auditiva tomando en cuenta las consideraciones de costo computacional.

Se determinó el desempeño de cada funcionalidad mediante distintos criterios.

Pruebas de uso que registren el desempeño y aprendizaje por parte de los usuarios constituyen una perspectiva de mejora para futuros trabajos.

X. REFERENCIAS BIBLIOGRÁFICAS

1. SCHAUERTE, Boris [et al.]. *An Assistive Vision System for the Blind That Helps Find Lost Things.* Alemania: Karlsruhe Institute of Technology, Institute for Anthropomatics, Adenauerring, 2012.

2. LI, Bing [et al.]. *Guided Text Spotting for Assistive Blind Navigation in Unfamiliar Indoor Environments.* Advances in Visual Computing, 12th International Symposium, ISVC 2016: 11-22.

3. SCHWARZE, T. [et al.]*. A Camera-based mobility aid for visually impaired people.* Kunstl Intell, 2015.

4. LI, Bing [et al.]. I*SANA: Wearable Context-Aware Indoor Assistive Navigation with Obstacle Avoidance for the Blind.* Nueva York: The City College, City University of New York, 2016.

5. LEE, Y.H.; MEDIONI, G. *Rgb-d camera based wearable navigation system for the visually impaired.* Los Ángeles, USA: Institute for Robotics and Intelligent Systems, University of Southern California, 2016.

6. *600 mil peruanos sufren discapacidad visual.* Diario la República [consultado el 18/08/2020]. Disponible en:

https://larepublica.pe/empresa/1267768-600-mil-peruanos-sufren-discapacidad-visual/.

7. *Unión Nacional de Ciegos del Perú.* UNCP, 2019 [consultado el 18/08/2020]. Disponible en: https://uncp.pe.

8. WEISBECKER, A. [et. al.]. *A survey: Outdoor Mobility Experiences by the Visually Impaired.* Mensch und Computer 2015 Workshopband, Stuttgart: Oldenbourg Wissenschaftsverlag, 2015: 391-397.

9. MANDAL, A. *What is visual impairment?.* News medical life sciences [consultado el 18/08/2020]. Disponible en: https://www.news-medical.net/health/What-is-visual-impairment.aspx.

10. *Sustitución sensorial: por qué la perspectiva ecológica podría ser la clave.* Open mind [consultado el 18/08/2020]. Disponible en:

https://www.bbvaopenmind.com/ciencia/biociencias/sustitucion-sensorial-la-perspectiva-ecologica-puede-la-clave/.

11. *Dispositivo de asistencia en la movilidad de personas ciegas y con baja visión* [consultado el 18/08/2020]. Disponible en: https://forms.gle/QuoocgM7nGay9Nrh6.

12. *BrainPort Vision Pro.* Wicab Inc. [consultado el 18/08/2020]. Disponible desde: https://www.wicab.com/brainport-vision-pro.

13. *Lazarillo GPS.* [consultado el 18/08/2020]. Disponible en: https://play.google.com/store/apps/details?id=com.lazarillo.

14. BHOWMICK, Alexy and HAZARIKA, Shyamanta M. An insight into assistive technology for the visually impaired and blind people: state-of-the-art and future trends. *Journal on Multimodal User Interfaces*. 2017. Vol. 11, no. 2p. 149–172. DOI 10.1007/s12193-016-0235-6.

15. Open CV tutorial. [consultado el 18/08/2020]. Disponible desde: https://opencvpythontutroals.readthedocs.io/en/latest/py_tutorials/py_calib3d/py_depthmap/py_depthmap.html

16. *Ceguera y discapacidad visual.* Organización Mundial de la Salud [consultado el 18/08/2020]. Disponible en:

https://www.who.int/es/news-room/fact-sheets/detail/blindness-and-visual-impairment.

17. *Datos de la ceguera y la discapacidad visual.* Organización Mundial de la Salud [consultado el 18/08/2020].

Disponible en:

https://www.who.int/features/factfiles/blindness/blindness_facts/es/

18. *Registro de personas con discapacidad visual en todo el Perú.* [consultado el 18/08/2020]. Disponible en:

https://www.conadisperu.gob.pe/notas-informativas/se-registran-801-mil-personas-con-discapacidad-visual-en-todo-el-peru