

Desarrollo de juego aplicando identificación de objetos con umbrales de colores

Torres Jonathan

Departamento de Ciencias de la Computación y Electrónica

UTPL

San Cayetano Alto, 1101608, Loja-Ecuador

jptorres6@utpl.edu.ec

Abstract—El desarrollo tecnológico en procesadores de computadores, cámaras digitales y gracias a librerías abiertas de procesamiento de imágenes (Open CV), a permitido que la visión artificial sea accesible a todo público. De esta manera se puede desarrollar soluciones a problemas de la sociedad.

En el presente documento se explicará el desarrollo de un programa capaz de identificar objetos por medio de su color y hacer que el usuario interactúe con un juego por medio un objeto. Para ello se utilizará el lenguaje de programación Python y las librerías Pygame y Open CV.

Index Terms—Visión artificial, Python, Open CV, Pygame, Procesamiento de Imágenes, Open Source, Juego.

I. INTRODUCCIÓN

LA visión artificial es una disciplina científica que incluye métodos para adquirir, procesar, analizar y comprender las imágenes del mundo real con el fin de producir información numérica o simbólica para que puedan ser procesados por un computador. Tal y como los humanos usamos nuestros ojos y cerebros para comprender el mundo que nos rodea, la visión por computador trata de producir el mismo efecto para que las computadoras puedan percibir y comprender una imagen o secuencia de imágenes y actuar según convenga en una determinada situación. La adquisición de los datos se consigue por medio de dispositivos como secuencias de imágenes, vistas desde varias cámaras de video o datos multidimensionales desde un escáner médico.

Este trabajo presenta el desarrollo de un juego para niños, con el objetivo de mejorar su movilidad, y de así el pueda hacer ejercicio mientras juega. Se utiliza el lenguaje de programación Python, que gracias a la cantidad de librerías que existen para este lenguaje se puede, procesar imágenes y realizar la interfaz gráfica para el juego. Para el procesamiento de imágenes se utiliza la librería Open CV que nos permite identificar objetos por medio de su color y así hacer que el computador identifique el objeto y su posición. Finalmente se realiza la interfaz gráfica se desarrolla a través de la librería Pygame.

II. MARCO TEÓRICO

Esta sección describe los conceptos más importantes acerca de los métodos usados para identificación de objetos, el lenguaje de programación Python y las librerías OpenCV y Pygame.

A. Detección de Objetos

La detección de objetos es la parte de la visión artificial que estudia cómo detectar la presencia de objetos en una imagen, bien sea atendiendo al tipo de objeto (una persona, un automóvil) o a la instancia del objeto. Generalmente se pueden distinguir dos partes en el proceso de detección: la extracción de características del contenido de una imagen y la búsqueda de objetos basada en dichas características[1].

Para ello vamos emplear la detección del objeto usando el espacio de color. A continuación se explicará y definirá cada una de las técnicas de segmentación de color existentes.

1) *Espacio RGB*: El espacio RGB es el espacio de color más extendido y el que utilizan la gran mayoría de cámaras de video y fotográficas para construir una imagen de color. Y de ahí, su importancia en visión artificial ya que trabajar con el mismo espacio de color con el que trabaja la cámara con la que se capturan las imágenes permite evitar la alteración de las propiedades del color durante el proceso de segmentación, propia de los errores de conversión y transformación, y por otro lado conseguir una mayor velocidad de segmentación por ahorro de esas operaciones de conversión y redondeo.

El espacio RGB se representa como un cubo dónde un color viene definido por la mezcla de valores de intensidad de tres colores primarios, rojo, verde y azul. Un color viene descrito por una tupla de 3-coordenadas en el cubo. El color negro se representa por ($r=0, g=0, b=0$) y el color blanco por ($r=255, g=255, b=255$). La gama acromática de escala de grises está representada por la diagonal del cubo[1].

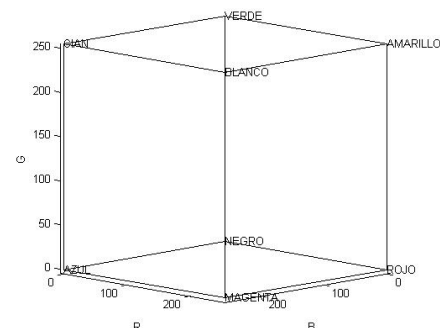


Fig. 1. Representación del espacio RGB

2) *Espacio HSV*: El espacio HSV representa uno de los espacios de coordenadas más clásicos e intuitivos existentes en la literatura. Su interpretación geométrica viene determinada por un cono de base quasi-hexagonal. Con esta representación, cada color trabaja con 3 componentes básicas: matiz, saturación y brillo. El matiz, hHSV, hace referencia al valor de cromaticidad o clase de color. La saturación, sHSV, se refiere a las longitudes de onda que se suman a la frecuencia del color, y determina la cantidad de blanco que contiene un color. Menos saturado este un color más cantidad de blanco, y contra más saturado este un color menor cantidad de blanco. En definitiva, la saturación representa la pureza e intensidad de un color. Así, la falta de saturación viene dada por la generatriz en la representación del cono HSV. Esa falta de saturación representa la gama de grises desde el blanco hasta el negro. La luminancia, vHSV, se corresponde con la apreciación subjetiva de claridad y oscuridad[2].

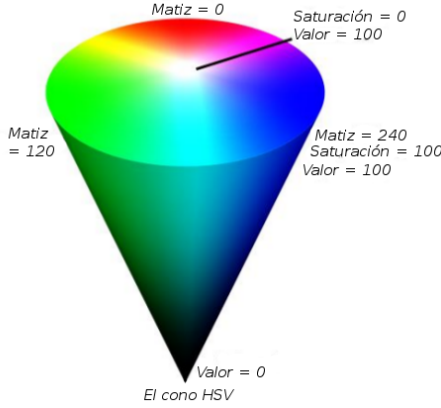


Fig. 2. Representación del espacio HSV.

Cuando se quiere representar una imagen a color en el espacio HSV, es importante determinar como influyen las componentes de color de una cámara de video RGB sobre el espacio HSV. Así, el sistema HSV viene definido por:

$$h_{HSV} = \begin{cases} \frac{g-b}{\max(r,g,b)-\min(r,g,b)} & \text{si } r = \max(r,g,b) \\ \frac{b-r}{\max(r,g,b)-\min(r,g,b)} + 2 & \text{si } g = \max(r,g,b) \\ \frac{r-g}{\max(r,g,b)-\min(r,g,b)} + 4 & \text{si } b = \max(r,g,b) \end{cases}$$

$$s_{HSV} = \frac{\max(r,g,b)-\min(r,g,b)}{\max(r,g,b)}$$

$$v_{HSV} = \max(r,g,b)$$
(1)

Donde g es la intensidad de verde, r la intensidad de rojo, b es la intensidad de azul.

Existen muchas variantes de espacios de color intuitivos, en función de cómo se modifique su representatividad, cabe destacar otros espacios como el HSI, HLS, y las variantes de éste último según su forma de construirlo[3].

El espacio de color HSV representa mejor que HLS la saturación, aunque tiene peor representación de la luminancia. El HLS viene a representarse como un doble cono donde los vértices determinan la máxima y mínima luminancia[3].

3) *HISTOGRAMA BIDIMENSIONAL V/S*: El histograma bidimensional V/S, es un histograma de dos variables cartesianas, luminancias y saturaciones. Su finalidad, radica en que proporciona un mecanismo sencillo e intuitivo para separar la información cromática de la acromática. La evaluación de un histograma V/S y la fijación de umbrales sobre él, permiten conseguir la extracción de reflejo y de distintos grados de sombras[4].

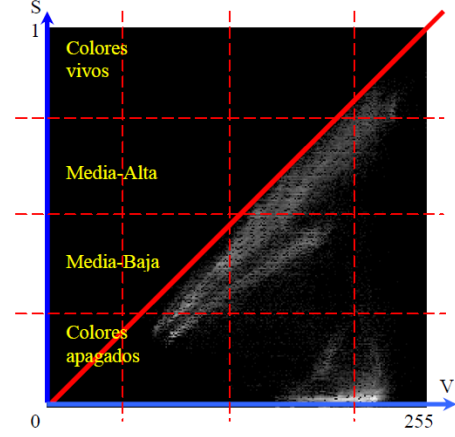


Fig. 3. Histograma Bidimensional V/S.

B. Métodos de segmentación

La segmentación empleada tanto con RGB como con HSV se fundamenta en la detección de umbrales del mapa de componentes RGB y/o HSV. La umbralización de imágenes de objetos en entornos robotizados muchas veces es compleja por la falta de conocimiento a priori del número de objetos a detectar, por la influencia de elementos no deseados como sombras, brillos, la complejidad de colores, texturas y tamaño de los objetos y la posibilidad de solapamiento de estos, así como las variaciones en el fondo. Esto hace que la elección de un número bajo de umbrales influya en la subsegmentación, se detectan menos regiones de las deseadas, o la elección de un número elevado lo haga en una sobresegmentación, se detectan más regiones de las que interesa. Es debido a la aparición de sombras y brillos por un lado, y por otro la similitud de los colores que tienen los objetos presentes en la escena, principalmente.

La umbralización de cada una de las componentes de color tanto RGB como HSV se ha llevado a cabo mediante la detección automática de multi-umbrales mediante el método de Otsu. Estimándose, el número mínimo de umbrales por componente en uno y el número máximo de umbrales en tres. El hecho, de que así se haya hecho es porque al menos en toda componente espacial de color es posible separar objeto de fondo, y para ello es necesario un umbral. Y el número máximo de tres viene determinado porque con tres umbrales por componente se tienen un máximo de 64 posibles clases/región que se consideran más que suficientes para segmentar una imagen. El número de clases/región distintas

viene determinado por:

$$nC = \prod_{k=1}^{nc} (nu_k + 1) \quad (2)$$

donde nc indica el número de capas en la imagen, que para las imágenes en color es siempre tres y corresponde a las componentes básicas: rojo, verde y azul en RGB o matiz, saturación y luminancia en HSV. Y nu_k hace referencia al número de umbrales detectados para cada capa de la imagen. Se ha optado por el método basado en información estadística y espacial de histogramas, como es el método de Otsu, fundamentado en distribuciones Gaussianas de partes del histograma. Y el cálculo de umbrales viene determinado por una aproximación para automatizar el método de Otsu[5].

C. Python

Python es un lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis que favorezca un código legible. Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, usa tipado dinámico y es multiplataforma. Es administrado por la Python Software Foundation. Posee una licencia de código abierto, denominada Python Software Foundation License.

El lenguaje de programación Python logra reconciliar evidentes contradicciones: es tan elegante como pragmático, es simple pero potente, es de muy alto nivel, aunque no resulta molesto cuando hay que modificar bits y bytes. Se adapta a programadores principiantes y a expertos[6].

D. Open CV

OpenCV es una biblioteca libre de visión artificial originalmente desarrollada por Intel. Desde que apareció su primera versión alfa en el mes de enero de 1999, se ha utilizado en infinidad de aplicaciones. Desde sistemas de seguridad con detección de movimiento, hasta aplicaciones de control de procesos donde se requiere reconocimiento de objetos. Esto se debe a que su publicación se da bajo licencia BSD, que permite que sea usada libremente para propósitos comerciales y de investigación con las condiciones en ella expresadas.

Open CV es multiplataforma, existiendo versiones para GNU/Linux, Mac OS X y Windows. Contiene más de 500 funciones que abarcan una gran gama de áreas en el proceso de visión, como reconocimiento de objetos (reconocimiento facial), calibración de cámaras, visión estereoa y visión robótica[7].

E. Pygame

Pygame es un conjunto de módulos del lenguaje Python que permiten la creación de videojuegos en dos dimensiones de una manera sencilla. Está orientado al manejo de sprites.

Gracias al lenguaje, se puede prototipar y desarrollar rápidamente. Esto se puede comprobar en las competiciones que se disputan online, donde es cada vez más usado. Los resultados pueden llegar a ser profesionales.

También puede utilizarse para crear otros programas multimedia o interfaces gráficas de usuario[8].

III. DESARROLLO

Para una mejor comprensión de como se desarrollo el juego se divide programa en: identificar el objeto por medio del color, juego, y juego detección de objetos.

A. Identificación objeto

Las librerías elegidas para el desarrollo del proyecto corresponden a “OpenCV 2.4.10”, el lenguaje de programación “Python 2.7” y el sistema operativo “Windows 10”. La técnica para el reconocimiento se basa en cinco pasos que se muestran a continuación:

- 1) Extraer la imagen de la cámara y guardarla en memoria, luego convertir la imagen del estándar RGB al modelo de color HSV.

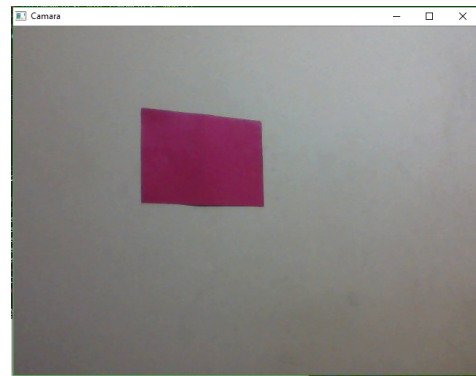


Fig. 4. Imagen de entrada en la cámara

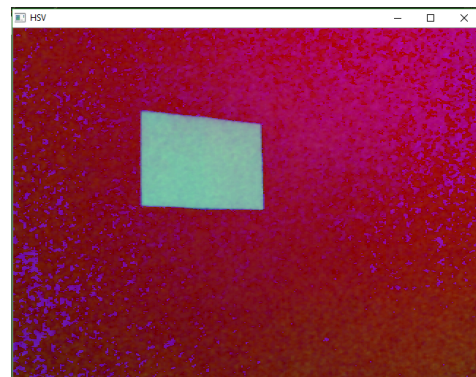


Fig. 5. Imagen de cámara en HSV

- 2) Aplicamos un desenfoque gaussiano que nos permita suavizar las características de la imagen.
- 3) Aplicar la operación de binarización con los valores de HSV acorde al color que se desea encontrar.

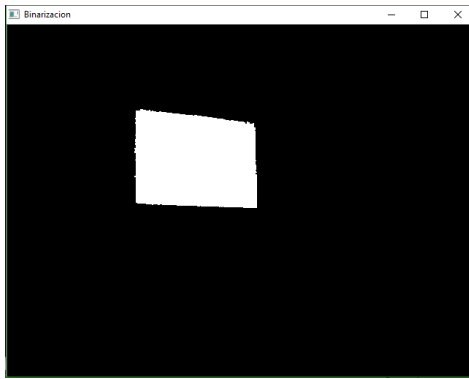


Fig. 6. Imagen HSV binarizada con enfoque en el color rojo.

- 4) Aplicar la operación morfológica de dilatación, de tal manera que ensanchemos un poco más los objetos que queremos identificar.

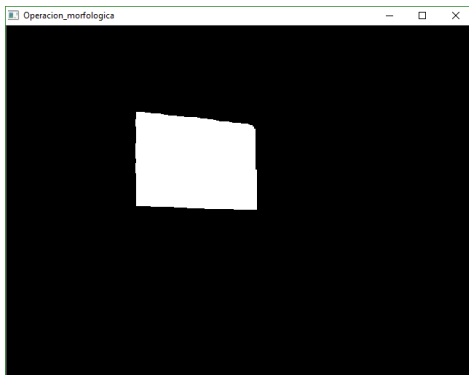


Fig. 7. Imagen binarizada dilatada para hacer más grandes los rasgos del objeto.

- 5) Encontrar el centro del objeto, y dibujarlo.

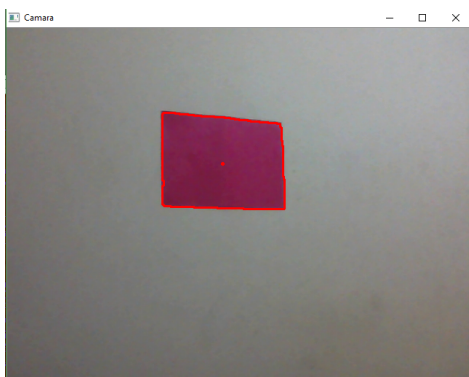


Fig. 8. Cento de objeto identificado.

B. Juego

El desarrollo de la interfaz se basa en el juego historico PACMAN, es un círculo amarillo al que le falta un sector, por lo que parece tener boca. Aparece en laberintos donde debe comer puntos pequeños (llamados «Pac-dots» en inglés), puntos mayores y otros premios con forma de frutas y otros objetos. El objetivo del personaje es comer todos los puntos

de la pantalla. La interfaz gráfica del juego se puede ver en la Figura.9

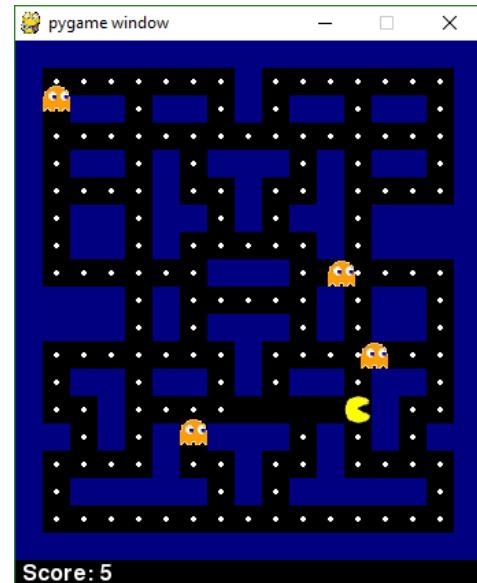


Fig. 9. Interfaz grafica del juego

C. Detección del objeto con juego

Para poder jugar se utilizaron tres colores de los cuales el usuario puede elegir(verde, amarillo, rojo). El usuario puede elegir el color del objeto con el cual va a jugar.



Fig. 10. Interfaz que le permite al usuario elegir el color del objeto para jugar.

Para inicia el juego y la interacción con los objetos se usa la siguiente interfaz(ver figura.11).

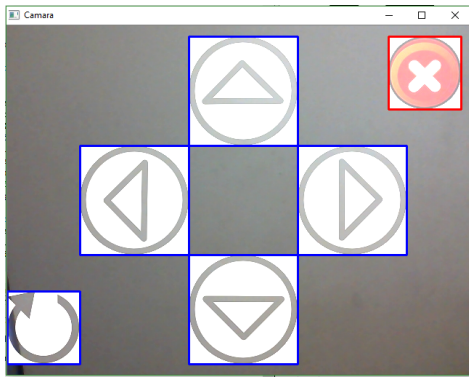


Fig. 11. Intefaz gráfica para interactuar con el juego por medio de la cámara.

Finalmente, se puede jugar(Ver Figura.12).

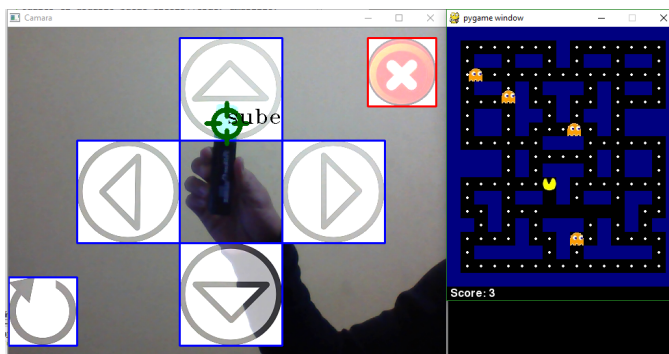


Fig. 12. Pantalla del juego.

IV. RESULTADOS

Para analizar los resultados se analiza a través de: umbrales de colores HSV y distancia máxima de identificación.

A. Umbrales de colores HSV

Normalmente en el espacio HSV, el eje horizontal del triángulo denota la saturación, mientras que el eje vertical corresponde al valor del color. De este modo, un color puede ser elegido al tomar primero el matiz de una región circular, y después seleccionar la saturación y el valor del color deseados de la región triangular.

El programa de se ha a decidido identificar 3 colores, los valores de matiz, saturación, brillo son los que están ubicados en la Tabla 1.

Color	Matiz	Saturación	Brillo
Verde bajo	49	50	50
Verde alto	100	255	210
Amarillo bajo	32	76	72
Amarillo alto	45	255	210
Rojo bajo	160	100	100
Rojo alto	179	255	255

Tabla.1. Valores altos y bajos de los colores a identificar en HSV(Matiz, saturación y brillo).

B. Distancia máxima de identificación

Para realizar las pruebas se utilizo un papel con las siguientes dimensiones:210 mm X 148 mm. Para identificar el objeto se calcula el área, esta debe ser mayor a 200000 $pixels^2$.

A continuación se mostrara los resultados obtenidos para distintas distancias(metros). En la tabla 2 se presenta los resultados a que distancia se pudo identificar a los objetos.

Distancia(m.)	Verde	Amarillo	Rojo
1 metro	Identifica	Identifica	Identifica
2 metros	Identifica	Identifica	Identifica
3 metros	Identifica	No identifica	Identifica
4 metros	Identifica	No identifica	No identifica
5 metros	No identifica	No identifica	No identifica

Tabla.2. Resultados de identificar el color a una cierta distancia.

V. CONCLUSIONES

El programa funciona sin fallas a una distancia máxima de 2 metros, a mayor distancia el programa falla en la identificación del objeto.

El lenguaje de programación Python, las librerías Open CV y Pygame, fueron muy útiles al momento de desarrollar el programa.

La cantidad de luz y objetos al fondo de la imagen pueden hacer que el programa falla, así que se recomienda usar un fondo de color blanco o negro.

Si al fondo de la imagen existen objetos de color similar al del objeto que deseamos identificar, estos objetos se tomarán como ruido el programa puede dar fallos.

REFERENCES

- [1] Castells, M. (2014). El impacto de internet en la sociedad: una perspectiva global. Recuperado de <https://goo.gl/V8hyIf>.
- [2] Smith, A.R., (1978). "Color gamut transformation pairs". Computer Graphics, Vol 12, No 3, pp 12-19.
- [3] Ortiz, F.G., (2002). "Procesamiento morfológico de imágenes en color. Aplicación a la reconstrucción geodésica". Tesis doctor de la Escuela Politécnica Superior. Universidad de Alicante. Disponible en: <https://goo.gl/F3yIe2>.
- [4] Angulo, J., (2003). "Morphologie mathématique at indexation d'images couleur. Application à la microscopie en biomedicine". Tesis doctor de la Ecole Des Mines de Paris.
- [5] Cheriet, M., Said, J.N, Suen, C.Y., (1998) "Arecursive thresholding technique for Image Segmentation", IEEE Transactions on Image Processing, Vol. 7, No 6, pp 918-921. Disponible en: <https://goo.gl/9SjzRR>.

- [6] Python Software Foundation. History and license.
Disponible en: [http://docs.python.org/3/
license.html](http://docs.python.org/3/license.html).
- [7] Open Source Initiative. The Open Source Definition.
Disponible en: [www.opensource.org/docs/
definition.php](http://www.opensource.org/docs/definition.php).
- [8] Sitio oficial de Pygame Disponible en: [http:
//www.pygame.org](http://www.pygame.org).