

# **RECONOCIMIENTO DE PELOTA Y SALTOS**

**JHONATAN PINEDA GÓMEZ**  
**JUAN JOSE VERA ARANGO**

**UNIVERSIDAD TECNOLÓGICA DE PEREIRA**  
**FACULTAD DE INGENIERÍAS**  
**INTELIGENCIA ARTIFICIAL**  
**PEREIRA**  
**2017**

## TABLA DE CONTENIDO

<b>RECONOCIMIENTO DE PELOTA Y SALTOS.....</b>	<b>3</b>
<b>VISIÓN POR COMPUTADOR.....</b>	<b>3</b>
<b>MODELO DE COLOR HSV .....</b>	<b>5</b>
<b>CARACTERÍSTICAS .....</b>	<b>6</b>
<b>Matiz.....</b>	<b>6</b>
<b>Saturación .....</b>	<b>8</b>
<b>Valor .....</b>	<b>8</b>
<b>TRANSFORMACIÓN DE RGB A HSV.....</b>	<b>9</b>
<b>PYTHON 3.5.....</b>	<b>10</b>
<b>NUMPY .....</b>	<b>10</b>
<b>OPENCV .....</b>	<b>11</b>
<b>Módulos de OpenCV.....</b>	<b>12</b>
<b>INSTALACIÓN DE LIBRERÍAS .....</b>	<b>13</b>
<b>DOCUMENTACIÓN.....</b>	<b>13</b>
<b>REFERENCIAS .....</b>	<b>15</b>

## **RECONOCIMIENTO DE PELOTA Y SALTOS**

### **VISIÓN POR COMPUTADOR**

La visión artificial o visión por computador 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 tratados 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. Esta comprensión se consigue gracias a distintos campos como la geometría, la estadística, la física y otras disciplinas. La adquisición de los datos se consigue por varios medios como secuencias de imágenes, vistas desde varias cámaras de video o datos multidimensionales desde un escáner médico.

Hay muchas tecnologías que utilizan la visión por computador, para desarrollar el software a presentar el enfoque será reconocimiento de objetos.

La detección de objetos es la parte de la visión artificial que estudia cómo detectar la presencia de objetos en una imagen sobre la base de su apariencia visual, bien sea atendiendo al tipo de objeto (una persona, un coche) o a la instancia del objeto (mi coche, el coche del vecino). 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.

La extracción de características consiste en la obtención de modelos matemáticos compactos que "resuman" el contenido de la imagen con el fin de simplificar el proceso de aprendizaje de los objetos a reconocer. Dichas características son comúnmente llamadas descriptores.

Existen diversos tipos de descriptores, que tendrán mejor o peor rendimiento en función al tipo de objeto a reconocer y a las condiciones del proceso de reconocimiento (la luz controlada o no, distancia al objeto a reconocer conocida o no). Se pueden usar desde básicos histogramas de color o intensidad de luz, descriptores LBP (Local Binary Pattern, usado sobre todo para texturas) o más avanzados como el HOG (Histogram of Oriented Gradients) o SIFT.

Para el proceso de clasificación se pueden usar diferentes técnicas de aprendizaje máquina. Existen diferentes métodos, como la regresión logística, o más avanzados basados en técnicas de aprendizaje automático como el SVM o AdaBoost (Adaptative Boost).

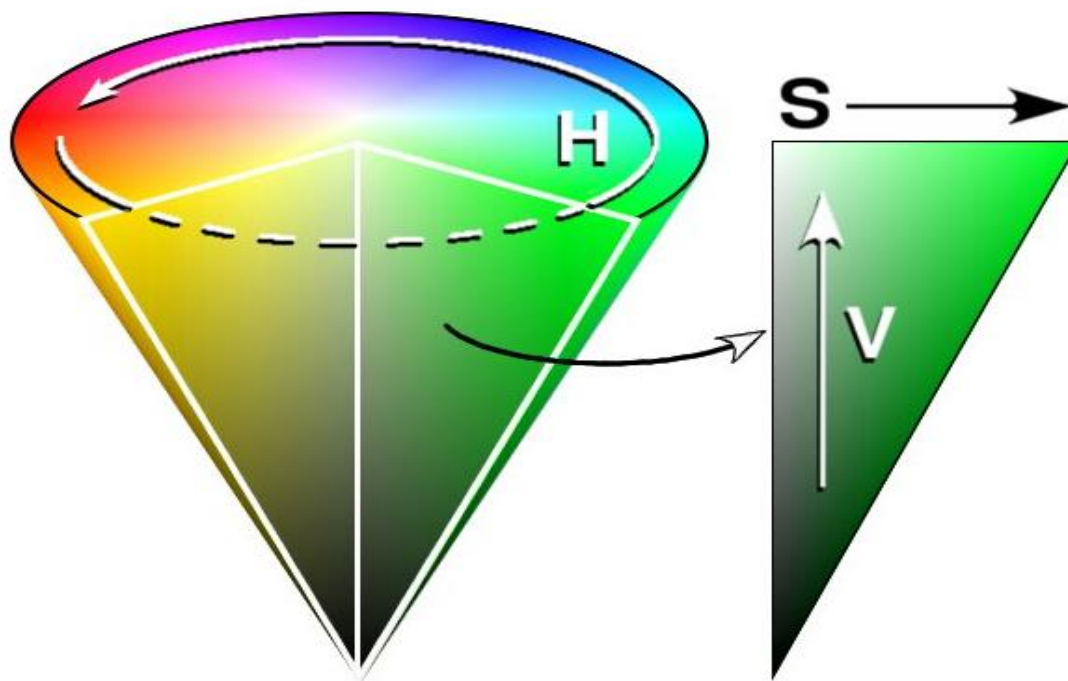
Teniendo claro el significado de visión por computador nos enfocaremos en el software a realizar que es el reconocimiento de una pelota y detectar la cantidad de saltos que esta realice.

## MODELO DE COLOR HSV

El modelo HSV (del inglés Hue, Saturation, Value – Matiz, Saturación, Valor)

Es común que deseemos elegir un color adecuado para alguna de nuestras aplicaciones, cuando es así resulta muy útil usar la ruleta de color HSV. En ella el matiz se representa por una región circular; una región triangular separada, puede ser usada para representar la saturación y el valor del color. Normalmente, 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 modelo HSV fue creado en 1978 por Alvy Ray Smith. Se trata de una transformación no lineal del espacio de color RGB,



## CARACTERÍSTICAS

### Matiz

Se representa como un grado de ángulo cuyos valores posibles van de 0 a 360° (aunque para algunas aplicaciones se normalizan del 0 al 100%). Cada valor corresponde a un color.

Ejemplos: 0 es rojo, 60 es amarillo y 120 es verde.

De forma intuitiva se puede realizar la siguiente transformación para conocer los valores básicos RGB: Disponemos de 360 grados dónde se dividen los 3 colores RGB, eso da un total de 120° por color, sabiendo esto podemos recordar que el 0 es rojo RGB(1, 0, 0), 120 es verde RGB(0, 1, 0) y 240 es azul RGB(0, 0, 1). Para colores mixtos se utilizan los grados intermedios, el amarillo, RGB(1, 1, 0) está entre rojo y verde, por lo tanto 60°. Se puede observar como se sigue la secuencia de sumar 60 grados y añadir un 1 o quitar el anterior:

Cono del modelo HSV.

$$0^\circ = \text{RGB}(1, 0, 0)$$

$$60^\circ = \text{RGB}(1, 1, 0)$$

$$120^\circ = \text{RGB}(0, 1, 0)$$

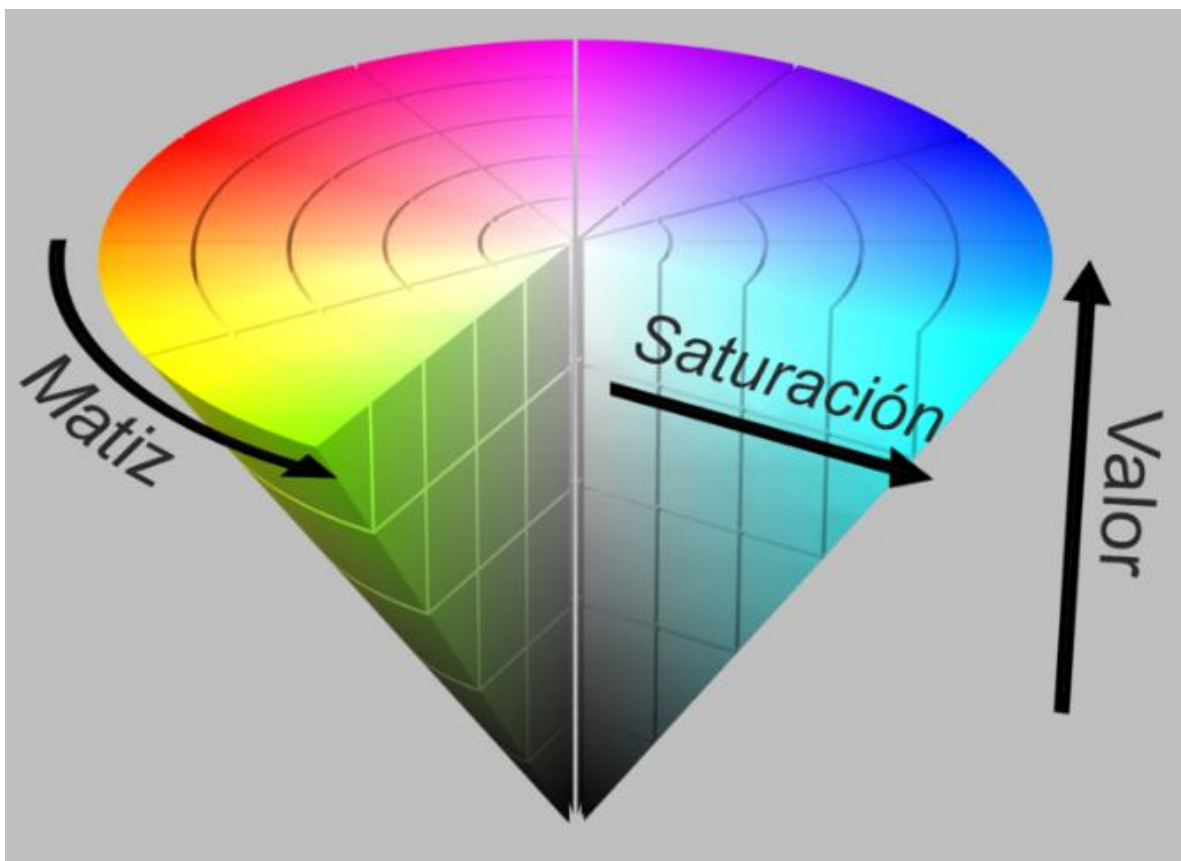
$$180^\circ = \text{RGB}(0, 1, 1)$$

$$240^\circ = \text{RGB}(0, 0, 1)$$

$$300^\circ = \text{RGB}(1, 0, 1)$$

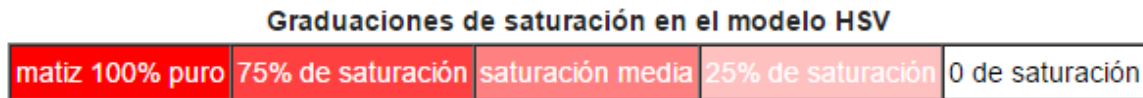
$$360^\circ = 0^\circ$$

Esta transformación permite saber los tonos de matices de colores puros que contienen alguna cantidad (o ninguna) de los colores R, G y B. Para el color blanco se puede poner cualquier color y saturación, siempre que se establezca el valor (de luminosidad) máximo. Asimismo, para el color negro se puede poner cualquier color y saturación, siempre que se ponga un valor de 0



## Saturación

Se representa como la distancia al eje de brillo negro-blanco. Los valores posibles van del 0 al 100%. A este parámetro también se le suele llamar "pureza" por la analogía con la pureza de excitación y la pureza colorimétrica de la colorimetría. Cuanto menor sea la saturación de un color, mayor tonalidad grisácea habrá y más decolorado estará. Por eso es útil definir la insaturación como la inversa cualitativa de la saturación.



## Valor

Representa la altura en el eje blanco-negro. Los valores posibles van del 0 al 100%. 0 siempre es negro. Dependiendo de la saturación, 100 podría ser blanco o un color más o menos saturado.



## TRANSFORMACIÓN DE RGB A HSV

Sea MAX el valor máximo de los componentes (R, G, B), y MIN el valor mínimo de esos mismos valores, los componentes del espacio HSV se pueden calcular como:

$$H = \begin{cases} \text{no definido,} & \text{si } MAX = MIN \\ 60^\circ \times \frac{G-B}{MAX-MIN} + 0^\circ, & \text{si } MAX = R \\ & \text{y } G \geq B \\ 60^\circ \times \frac{G-B}{MAX-MIN} + 360^\circ, & \text{si } MAX = R \\ & \text{y } G < B \\ 60^\circ \times \frac{B-R}{MAX-MIN} + 120^\circ, & \text{si } MAX = G \\ 60^\circ \times \frac{R-G}{MAX-MIN} + 240^\circ, & \text{si } MAX = B \end{cases}$$
$$S = \begin{cases} 0, & \text{si } MAX = 0 \\ 1 - \frac{MIN}{MAX}, & \text{en otro caso} \end{cases}$$
$$V = MAX$$

## PYTHON 3.5

### NUMPY

NumPy es una extensión de Python, que le agrega mayor soporte para vectores y matrices, constituyendo una biblioteca de funciones matemáticas de alto nivel para operar con esos vectores o matrices. El ancestro de NumPy, Numeric, fue creado originalmente por Jim Hugunin con algunas contribuciones de otros desarrolladores. En 2005, Travis Oliphant creó NumPy incorporando características de Numarray en NumPy con algunas modificaciones.

NumPy es el paquete fundamental para la computación científica con Python. Contiene entre otras cosas:

- un poderoso N-dimensional objeto de matriz sofisticados (radiodifusión) funciones
- herramientas para la integración de C / C ++ y Fortran
- álgebra lineal útil, transformada de Fourier, y capacidades de números aleatorios

Además de sus usos científicos obvias, NumPy también se puede utilizar como un contenedor multidimensional eficiente de los datos genéricos. Tipos de datos arbitrarias pueden ser definidos. Esto permite NumPy para integrar sin problemas y rápidamente con una amplia variedad de bases de datos.

## **OPENCV**

OpenCV es una librería de visión por computador de código abierto, La librería está escrita en los lenguajes C y C++ y es compatible con Linux, Windows y Mac OS X. Cuenta con un desarrollo activo en interfaces para Python, Ruby, Matlab y otros lenguajes.

OpenCV ha sido diseñado para ser eficiente en cuanto a gasto de recursos computacionales y con un enfoque hacia las aplicaciones de tiempo real.

OpenCV está escrito y optimizado en C y puede tomar ventaja de los procesadores con

Múltiples núcleos. Uno de los objetivos de OpenCV es proveer una infraestructura de visión por computador fácil de utilizar que ayuda a los programadores a desarrollar aplicaciones ‘satisfechas’ de CV rápidamente. La librería OpenCV contiene aproximadamente 500 funciones que abarcan muchas áreas de CV, incluyendo inspección de productos de fábricas, escaneo médico, seguridad, interfaces de usuario, calibración de cámaras, robótica...etc, porque la visión por computador y el aprendizaje automático van de la mano. OpenCV también tiene una completa librería de uso general de aprendizaje automático (MLL o Machine Learning Library), la cual es muy útil para cualquier problema de aprendizaje automático. Esta sublibrería está especializada en el reconocimiento estadístico de patrones y clustering.

## Módulos de OpenCV

OpenCV tiene una *estructura modular*. Los módulos principales de OpenCV los listo debajo.

### ***–core:***

Este es el módulo básico de OpenCV. Incluye las estructuras de datos básicas y las funciones básicas de procesamiento de imágenes. Este módulo también es usado por otros módulos como highgui.

### ***–highgui:***

Este módulo provee interfaz de usuario, códecs de imagen y vídeo y capacidad para capturar imágenes y vídeo, además de otras capacidades como la de capturar eventos del ratón...etc. Si necesitas capacidades de UI (User Interface) más avanzadas debes usar frameworks tales como Qt, WinForms...etc.

### ***–imgproc:***

Este módulo incluye algoritmos básicos de procesado de imágenes, incluyendo filtrado de imágenes, transformado de imágenes...etc.

### ***–video:***

Este módulo de análisis de vídeo incluye algoritmos de seguimiento de objetos, entre otros...

### ***–objdetect:***

Incluye algoritmos de detección y reconocimiento de objetos para objetos estándar.

## INSTALACIÓN DE LIBRERÍAS



[Docs](#) » [OpenCV-Python Tutorials](#) » [Introduction to OpenCV](#) » [Install OpenCV-Python in Windows](#)  
[Edit on GitHub](#)

### Install OpenCV-Python in Windows

#### Goals

In this tutorial

- We will learn to setup OpenCV-Python in your Windows system.

*Below steps are tested in a Windows 7-64 bit machine with Visual Studio 2010 and Visual Studio 2012. The screenshots shows VS2012.*

## Obtaining NumPy & SciPy libraries

See also:

[Installing the SciPy Stack](#)

### Official Source and Binary Releases


For each official release of NumPy and SciPy, we provide source code (tarball) as well as binary wheels for several major platforms. Binary wheels for SciPy on Windows are not available, we strongly recommend using a scientific Python distribution (those all do provide SciPy).

Project	Available packages	Download location
NumPy	Official <i>source code</i> (all platforms) and <i>binaries</i> for Windows, Linux and Mac OS X	<a href="#">PyPI page for NumPy</a>

## DOCUMENTACIÓN

### OpenCV-Python Tutorials


- [Introduction to OpenCV](#)



Learn how to setup OpenCV-Python on your computer!

- [Gui Features in OpenCV](#)

Here you will learn how to display and save images and videos, control mouse events and create tra



- [Core Operations](#)

In this section you will learn basic operations on image like pixel editing, geometric transformations

- [Image Processing in OpenCV](#)



In this section you will learn different image processing functions inside OpenCV.

- [Camera Calibration and 3D Reconstruction](#)



In this section we will learn about camera calibration, stereo imaging etc.

- [Machine Learning](#)



In this section you will learn different image processing functions inside OpenCV.

- [Computational Photography](#)

In this section you will learn different computational photography techniques like image denoising e

- [Object Detection](#)



In this section you will object detection techniques like face detection etc.

- [OpenCV-Python Bindings](#)



In this section, we will see how OpenCV-Python bindings are generated

## REFERENCIAS

<http://www.numpy.org/>

<http://sourceforge.net/projects/opencvlibrary/>

[https://opencv-python-tutroals.readthedocs.io/en/latest/py\\_tutorials/py\\_tutorials.html](https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_tutorials.html)

<http://opencvjaveriana.wikispaces.com/>

[https://opencv-python-tutroals.readthedocs.io/en/latest/py\\_tutorials/py\\_setup/py\\_table of contents setup/py table of contents setup.html#py-table-of-content-setup](https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_setup/py_table_of_contents_setup/py_table_of_contents_setup.html#py-table-of-content-setup)

<https://osl.ull.es/software-libre/opencv-libreria-vision-computador/>

[https://opencv-python-tutroals.readthedocs.io/en/latest/py\\_tutorials/py\\_setup/py setup in windows/py setup in windows.html#install-opencv-python-in-windows](https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_setup/py_setup_in_windows/py_setup_in_windows.html#install-opencv-python-in-windows)

<https://www.scipy.org/scipylib/download.html>