

# Uso de filtro de Kalman para rastreo de piel

Ulises Jiménez Guerrero<sup>1</sup>

<sup>1</sup>Instituto de Investigaciones en Inteligencia Artificial, Universidad Veracruzana, Xalapa, Veracruz, México

## Introducción

Este proyecto consiste de dos partes: detección de piel en imágenes mediante modelos probabilistas en el espacio de color, y seguimiento del movimiento mediante el uso de filtros de Kalman. Se desarrolló en el lenguaje de programación Python, con el uso de las paquetes de Numpy, SciPy y OpenCV. Ambas partes del proyecto se combinaron para su uso en tiempo real mediante una webcam. El programa desarrollado tiene la licencia GNU versión 3 y se encuentra disponible en <https://github.com/Ulisex666/FiltroKalmanPiel>

## Detección de tonos de piel en imágenes

Se tomaron muestras de píxeles de piel y de fondo en imágenes de autoría propia, y se calculó el valor promedio y la matriz de covarianza [fig. 1]. Con estos datos, se utilizó SciPy para generar una distribución de probabilidad normal multivariada para los tonos de piel y otra para los colores de fondo. Para la inferencia se utiliza el Teorema de Bayes. Para un píxel  $x$  se tiene

$$\Pr(x|w) = \text{Norm}_x[\mu_w, \Sigma_w],$$

donde  $w \in \{\text{Piel, Fondo}\}$ . Con una variación del Teorema de Bayes, se sigue que:

$$P(w|x) \propto P(x|w)P(w).$$

Como no se tiene información *a priori* del ambiente, se toma  $P(w) = 0,5$  para ambos casos posibles. De esta forma, se evalúa la probabilidad de que cada píxel sea piel o fondo, y se le asigna la etiqueta con el mayor valor de entre los dos. En este caso, se trabaja en el espacio de color HSV, tomando solamente los canales de *hue* y *saturation*, los componentes cromáticos del píxel.

## Filtro de Kalman

El filtro de Kalman, desarrollado por Rudolf E. Kálmán, es un algoritmo que permite estimar el estado de un sistema a través del tiempo, utilizando medidas ruidosas. Tiene la capacidad de estimar variables no observadas o no medidas. Los resultados que obtiene son más precisos que aquellos dados por los sensores con ruido. Se tienen dos suposiciones: el sistema a modelar es lineal y que nuestras mediciones son ruidosas siguiendo una distribución normal. Se siguen dos procesos principales: la fase de predicción y la fase de corrección. Este algoritmo es ampliamente usado para rastreo de objetos, en sistemas de navegación y en control.

## Resultados

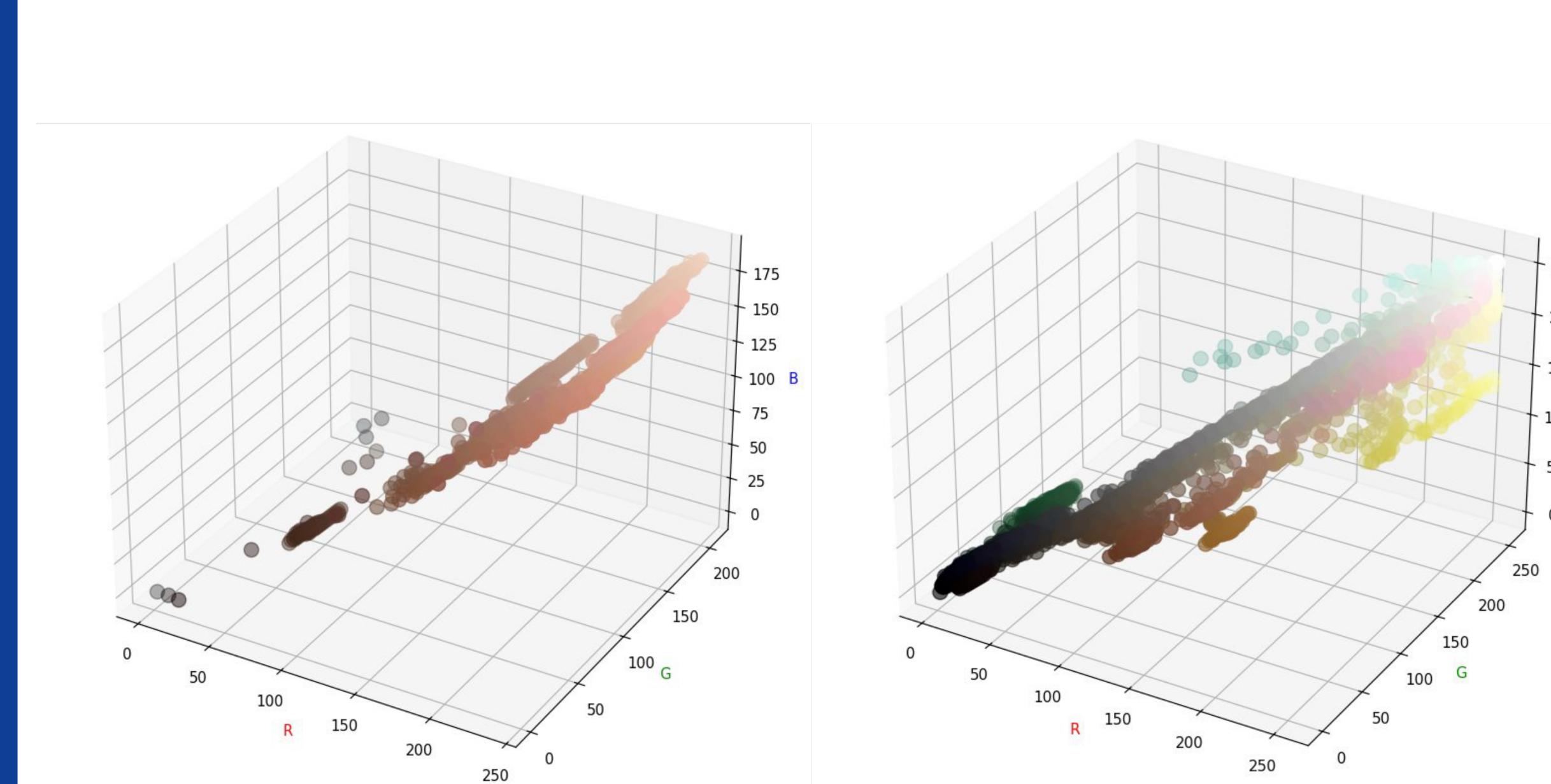


Figura 1: Colores utilizados para la creación de las distribuciones de probabilidad para piel y para fondo, respectivamente.



Figura 2: Resultados de la detección de piel mediante métodos probabilísticos en una imagen de prueba. Dada su naturaleza estática, se pudo utilizar una probabilidad a priori para la piel de 0.40.

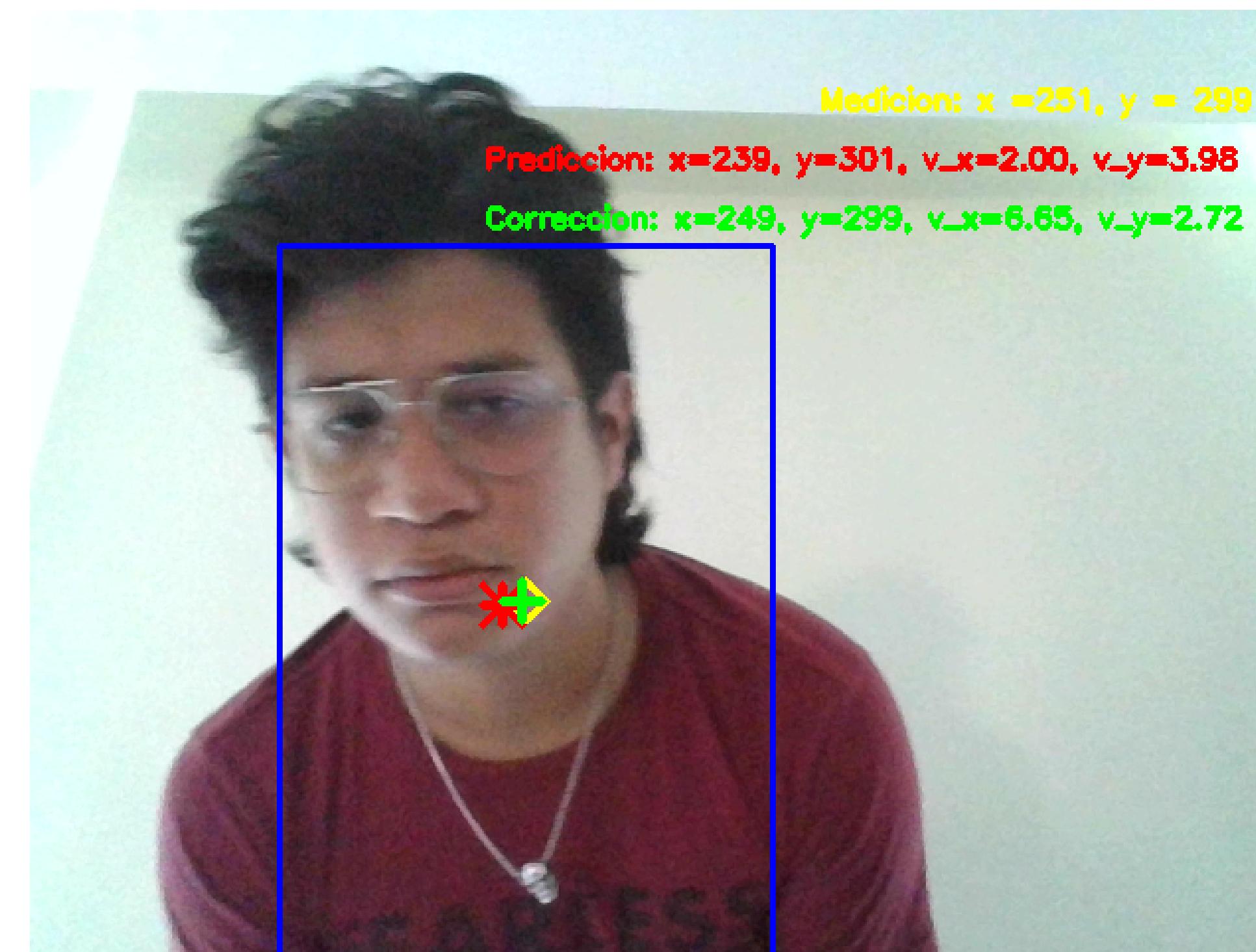


Figura 3: Programa de rastreo de piel en ejecución. Se observa la sección detectada como piel, la medición del centro, la predicción del algoritmo y la medición corregida, cada una con un indicador de diferente color.



Figura 4: Mediante el uso de las mediciones, el programa es capaz de corregir rápidamente la discordancia en las predicciones. Podemos observar como la predicción, marcada con rojo, es corregida y sigue de cerca a las mediciones. Esta corrección se dio en dos frames.

## Fases del filtro de Kalman

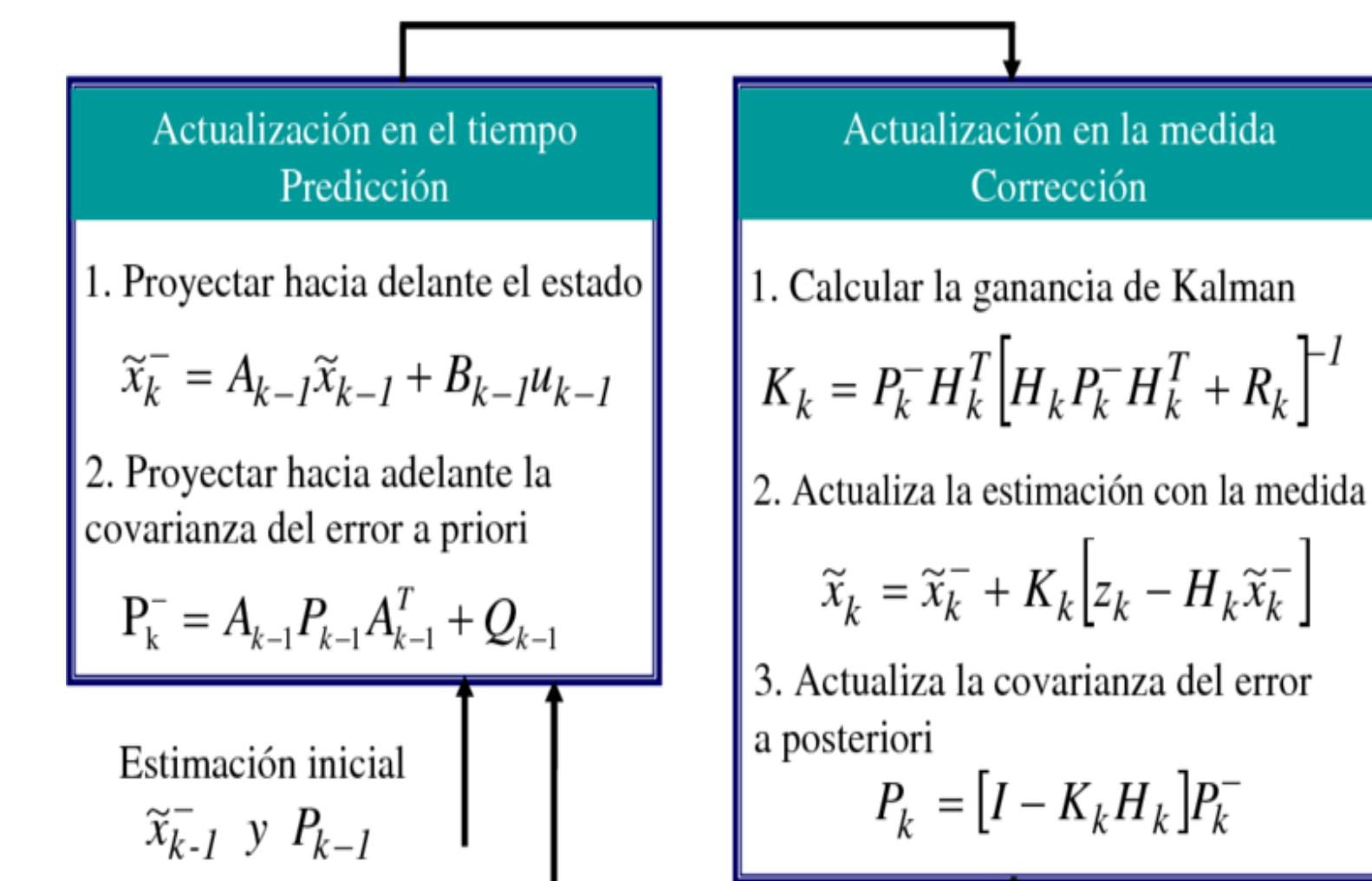


Figura 5: Diagrama del ciclo principal en el filtro de Kalman

El funcionamiento del filtro de Kalman se basa en la actualización del estado del sistema  $x$  en el tiempo  $k$  utilizando la matriz de transición  $A$  y una posible matriz de control  $B$ . A su vez, se actualiza la matriz de covarianza para el error  $P$  utilizando la misma matriz de transición  $A$  y el ruido modelado con una matriz de covarianza  $Q$ . Para la corrección, se tiene la ganancia de Kalman  $K$ . Mientras más grande sea su valor, más confianza se tiene en las mediciones  $z$  comparadas con las predicciones  $x$ . A su vez, se tiene la matriz  $H$  que relaciona las mediciones con el estado del sistema. Finalmente, se actualiza la covarianza del error y la estimación con base a las mediciones, y estos resultados se vuelven a utilizar para la siguiente predicción.

## Implementación del algoritmo

El programa funciona en tiempo real, usando la webcam del dispositivo. Utilizando el programa para detección de piel, se crea un polígono alrededor de los píxeles detectados como piel, y se busca rastrear el movimiento del centro de esta figura. Para esto, se genera un objeto filtro de Kalman mediante OpenCV.

Este es alimentado con las matrices de transición [fig. 6], de mediciones, de ruido del proceso, ruido de la medición y covarianza del error adecuadas, tomadas de la literatura. Con esto, para la primera iteración se da como estado inicial el centro de la imagen. El algoritmo procede prediciendo la ubicación del centro, midiéndola y posteriormente corrigiendo la predicción. Estos tres resultados son visibles en tiempo real.

Dadas las características del filtro de Kalman, se pueden estimar parámetros que no se observan directamente. En nuestro caso, estos son la velocidad en el eje X y en el eje Y de la imagen.

$$\begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Figura 6: Matriz de transición para movimiento en dos dimensiones. Cada columna indica: posición en X, posición en Y, velocidad en X y velocidad en Y.

## Bibliografía

- [1] Tim Babb. How a Kalman filter works, in pictures | Bzarg, August 2015.
- [2] Gary R. Bradski and Adrian Kaehler. Learning OpenCV: computer vision with the OpenCV library. EBL-Schweitzer. O'Reilly, Beijing Köln, 2008.
- [3] Joris Schutter, Jan Geeter, Tine Lefebvre, and Herman Bruyninckx. Kalman Filters: A Tutorial. Journal A, 44(4):52–59, 1999.