

# INSTITUTO TECNOLÓGICO DE BUENOS AIRES

22.05 ANÁLISIS DE SEÑALES Y SISTEMAS DIGITALES

---

## Trabajo práctico N°4

Entrega N°1

---

### *Grupo 3*

MECHOULAM, Alan	58438
LAMBERTUCCI, Guido Enrique	58009
RODRIGUEZ TURCO, Martín Sebastian	56629
LONDERO BONAPARTE, Tomás Guillermo	58150

### *Profesores*

Jacoby, Daniel Andres  
Belaustegui Goitia, Carlos F.  
Iribarren, Rodrigo Iñaki

Presentado: 17/06/20

*En el siguiente trabajo se presenta el estudio, investigación y análisis de un proceso de seguimiento del movimiento de un objeto en tiempo real, siendo conocida su posición inicial. Se realizaron pruebas tanto con videos como con cámaras web, pudiendo alternar el modo de trabajo.*

## I. INTRODUCCIÓN

Una imagen puede ser interpretada como una función bidimensional  $f(x, y)$ , donde tanto  $x$  como  $y$  representan en un plano el espacio visualizado, mientras que la misma función  $f(x, y)$  es la intensidad de la imagen bajo un punto dado. Cuando  $x$ ,  $y$  y  $f(x, y)$  son valores cuantizados y discretizados, la imagen se transforma en una imagen digital.

El procesamiento de dichas se define como el conjunto de técnicas aplicadas a estas imágenes, con el objetivo extraer información de ellas. Estas actividades cubren un campo que abarca un sin fin de aplicaciones, ya que se vale de máquinas capaces de detectar la totalidad del espectro electromagnético. Esto significa que se pueden obtener imágenes generadas por fuentes que captan información la cual para los humanos no se asocian con imágenes propiamente dichas, como lo son las ondas de radio, entre tantas otras.

Es posible considerar tres tipos de procesos computarizados en el procesamiento de imágenes, basándose en el nivel de tratamiento que se aplique, siendo así clasificados en bajo, medio y alto nivel. Los primeros incluyen actividades tales como reducción de ruido y aumento de contraste, tareas caracterizadas por el hecho de que tanto la entrada como la salida son imágenes. Las actividades de medio nivel de procesamiento incluyen trabajos de segmentación, es decir, identificar regiones u objetos dentro de las imágenes, descripción y clasificación de dichos elementos. Es así que esta categoría es destacada por sus salidas, ya que suelen ser información extraída de las imágenes a la entrada. Por último, los procesos de alto nivel se caracterizan por no solo reconocer objetos y analizarlos, sino también por darles un tratamiento normalmente asociado con la visión, tales así como “darles sentido” [1].

## II. INVESTIGACIÓN

Dada una señal continua a la entrada del sistema, una imagen sufre de dos procesos claves: **cuantización** y **discretización**. Si bien ambos refieren a tomar variables continuas y almacenarlas en memoria como variables discretas, se realiza esta diferenciación entre ambas ya que la primera hace referencia a la amplitud de la señal mientras que la segunda a coordenadas, que para el caso del estudio de imágenes, se refiere a píxeles, siendo estos el mínimo elemento que compone una imagen digital.

Ejemplificando lo anterior, se toma una entrada al sistema, como puede ser la presentada en la Figura (1), la cual, como ya se ha mencionado, es continua en  $x$ ,  $y$  y  $f(x, y)$ .



Figura 1: Entrada continua al sistema.

Por lo tanto, se deben tomar coordenadas finitas, por ejemplo, aquellas que se encuentran sobre la recta AB, y asignarle a cada una un valor dado de amplitud. En la Figura (2) se observa como una recta continua paralela al eje  $x$  (horizontal), la cual posee ciertas variaciones aleatorias dadas por el ruido existente, es dividida en una cierta cantidad de posiciones equiespaciadas (discretización), marcadas con cuadrados blancos sobre la curva, asignándoles un nivel específico en la escala de grises (cuantización), marcado con una línea negra por la izquierda de dicha escala.

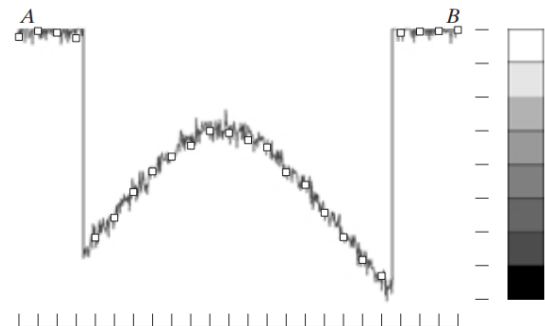


Figura 2: Amplitud de la escala de grises en la recta AB y muestreo de valores.

Realizando el mismo proceso para todos los niveles de discretización en el eje  $y$  (vertical), se obtiene finalmente una imagen digitalizada, la cual se la compara a continuación con la original [2].

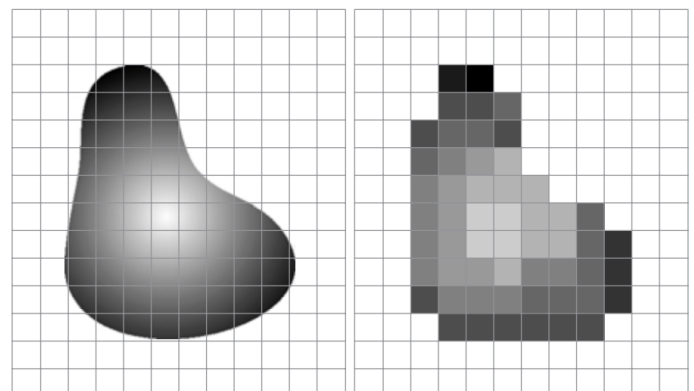


Figura 3: Imagen original comparada con la imagen digitalizada a procesar.

- Descripción de los conceptos más importantes de cada una.
- Análisis propio de lo presentado.
- Simulaciones de lo más relevante (códigos como apéndice).
- Elección del camino y justificación.

Ehh... es necesario estos 4 puntos que quedaron? Yo pondría solo el código o algo así.

Este trabajo se centra en procesos de medio nivel. Dicha definición es muy amplia, por lo cual es necesario acotar este camino. Es por ello que se decidió centrarse en el seguimiento de objetos en imágenes en movimiento. Se busca que, dada ciertas condiciones iniciales conocidas (brindadas por el usuario) en una imagen en movimiento en tiempo real, tomar un conjunto de datos de  $x$ ,  $y$  y  $f(x, y)$  para así seleccionar un elemento y seguir su trayectoria a través del tiempo, como hipótesis plantearemos:

- El objeto no cambiará rápidamente de color ni su iluminación o exposición-
- El objeto no sufrirá una oclusión por parte de un obstáculo del mismo color.

para lograr este fin se utilizó el algoritmo de Lucas-Kanade de Optical Flow, el método de Shi-Tomasi, la utilización de filtros de Kalman y filtrado de color.

#### II-A. Optical Flow

El campo de movimiento de una imagen es el movimiento real del objeto en el espacio proyectado sobre el plano de la imagen. El Optical Flow (flujo óptico) se define como el flujo de la intensidad en escala de grises en el plano de la imagen, a medida que evoluciona en el tiempo. También se puede interpretar el Optical Flow o flujo de la imagen como el movimiento aparente de la imagen basado en la percepción visual, y tiene dimensión de velocidad  $\vec{V} = (V_x, V_y)$ . Si el Optical Flow se determina de dos imágenes consecutivas, aparece un vector de desplazamiento  $\vec{d}$  de las cualidades elegidas, entre el cuadro  $n$  y el  $n+1$ .



(a) first frame (b) second frame (c) optical flow

Figura 4: Cubo de rubik rotando en una mesa.

Para la implementación del algoritmo existen varios caminos, como el basado en cálculo de gradientes de la imagen total, existe otro que utiliza solo ciertos puntos que se determinan constantes de la imagen, nosotros utilizaremos el método piramidal de Lucas-Kanade.

*II-A1. Lucas-Kanade:* Este consta del uso de información obtenida a partir de la intensidad del gradiente espacial para buscar la posición que mejor se acomoda a una imagen en movimiento [4] [5], algunas de las hipótesis que postula Lucas-Kanade son:

- Los movimientos entre cuadros consecutivos son pequeños. Tan pequeños como un pixel.
- La intensidad de los objetos se mantiene constante cuadro a cuadro.

Este algoritmo se basa en el principio de "Divide y conquistarás" al realizar la tarea de detectar movimiento en toda la imagen en problemas mas sencillos, consta de dividir la pantalla en un árbol cuaternario, y calculando recursivamente el optic flow de cada partición.

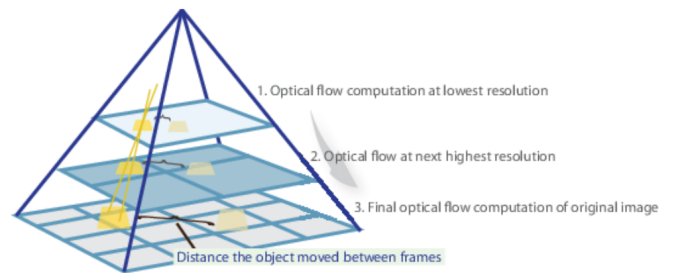


Figura 5: Cálculo del Optical Flow.

#### II-B. Filtro de Kalman

El filtro de Kalman es un filtro recursivo que busca estimar el estado de un sistema dinámico lineal discretizado de dimensión  $n$  mediante una serie de mediciones con ruido a partir de la descripción del modelo físico que rige las variables del vector de estado [6], [7].

*II-B1. Planteo del problema:* Dado un proceso estocástico lineal en tiempo discreto definido por

$$x_k = Ax_{k-1} + w_{k-1}^1 \quad (1)$$

donde  $x_k \in \mathcal{R}^N$  y siendo una medición en tiempo  $k$  definida por

$$z_k = Hx_k + v_k \quad (2)$$

con  $z_k \in \mathcal{R}^m$ , donde  $w_k$  y  $v_k$  son variables aleatorias que representan el ruido del proceso y de observación respectivamente, las cuales se asumen que son independientes entre sí y con distribuciones de probabilidad normales multivariadas definidas como

$$p(w) \sim N(0, Q)$$

$$p(v) \sim N(0, R)$$

donde  $Q$  y  $R$  son las matrices de covarianza del ruido del proceso y ruido de medición respectivamente, las cuales se

<sup>1</sup>Se omitió por simplicidad la función de control.

asumen constantes; donde la matriz de transición de estados  $A$  de dimensión  $n \times n$  —siendo  $n$  la dimensión de estados dinámicos del modelo— define la relación entre entados del paso  $k - 1$  al paso  $k$  sin contar el ruido del proceso; donde la matriz de transición de medición  $H$  de dimensión  $m \times n$  —siendo  $m$  la dimensión del vector de medición— fija la relación entre la medición realizada y el vector de estados, sin contar el ruido de medición; y finalmente donde el estado del filtro puede representarse como  $\hat{x}_{k|k}$  y  $P_{k|k}$ , siendo estas el estimador del vector de estados a posteriori dadas las mediciones hasta un tiempo  $k$ , y el estimador de la matriz de covarianza a posteriori dadas las mediciones hasta un tiempo  $k$  respectivamente. Se puede definir la operación del filtro de Kalman separando a esta en dos fases: la predicción, y la corrección.

#### II-B2. Ecuaciones de predicción:

$$\hat{x}_{k|k-1} = A\hat{x}_{k-1|k-1} \quad (3)$$

$$P_{k|k-1} = AP_{k-1|k-1}A^T + Q \quad (4)$$

#### II-B3. Ecuaciones de observación:

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k y_k \quad (5)$$

$$P_{k|k} = (\mathbb{I} - K_k H) P_{k|k-1} \quad (6)$$

donde,

$$y_k = z_k - H\hat{x}_{k|k-1} \quad (7)$$

$$S_k = HP_{k|k-1}H^T + R_k \quad (8)$$

$$K_k = P_{k|k-1}H^T S_k^{-1} \quad (9)$$

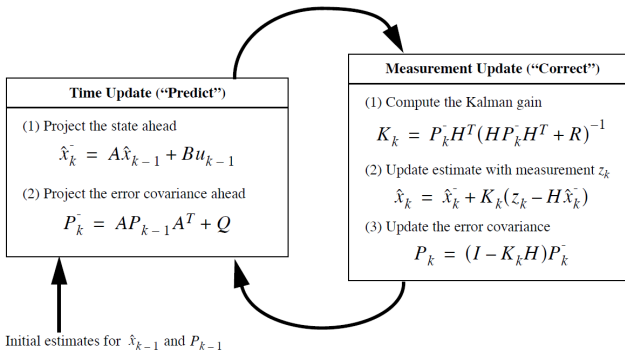


Figura 6: Funcionamiento completo del filtro Kalman simple [7].

En la Figura (7) se puede observar el funcionamiento del algoritmo con un vector de estados dinámicos de dimensión dos compuesto por las coordenadas  $(x, y)$  sobre el plano cartesiano, realizando mediciones periódicamente. Se comparan dos curvas. Por un lado, una senoidal con un ruido gaussiano montado sobre ella, simulando una serie de mediciones con ruido, y por el otro lado, la estimación obtenida a partir del filtro desarrollado.

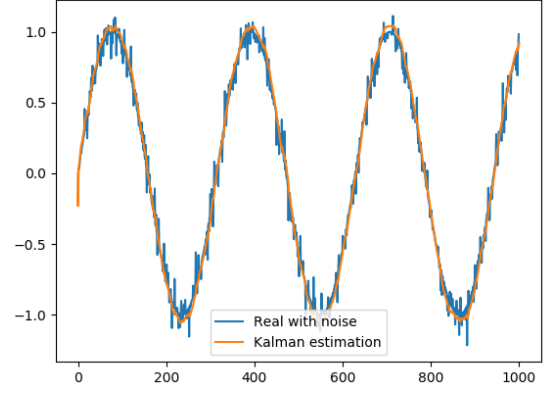


Figura 7: Seno con ruido comparada con estimación de Kalman.

**II-B4. Filtro de Color:** Se utilizó una máscara en todos los frames en base al color promedio de la selección inicial del usuario, el color promedio se obtiene al calcular la mediana de la selección.

### III. APORTES

Se decidió implementar código en **Python**, apoyándose en la librería **OpenCV**.

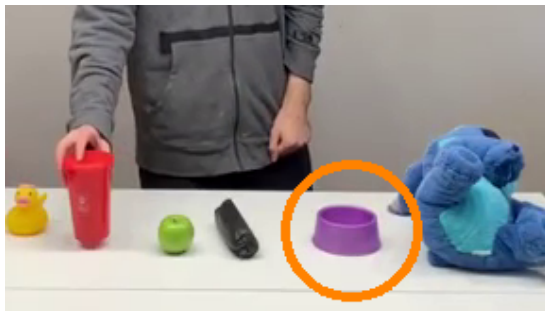
Primero se desarrolló un programa capaz de realizar la estimación de una variable mediante el uso del filtro de Kalman.

#### III-A. Implementación

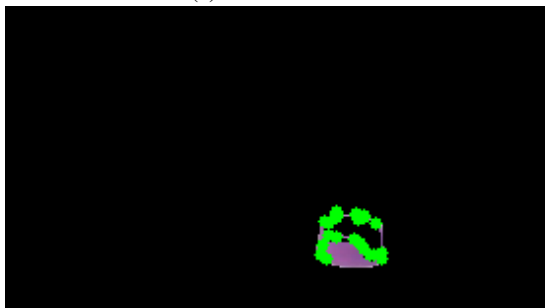
A modo de resumen revisaremos como se encuentra hasta el momento el tracker. A partir de la selección inicial se obtienen las mejores features del objeto seleccionado, luego mediante el algoritmo de Lucas-Kanade de Optical Flow se obtiene la evolución de las features en el tiempo, luego a este cúmulo de puntos se le calcula la media (centro de masas) y el desvío (centro de inercia), además se rechazan los puntos que se encuentren a una distancia mayor a un  $\sigma$  respecto al centro de masas, luego este centro de masas medido es el que se introduce en el filtro de Kalman para obtener una predicción. Este algoritmo cuenta con un problema que es el siguiente: Dados dos objetos A y B, donde A es el objeto a trackear, en el caso de que se crucen A y B esta configuración podría dejar de seguir a A y comenzar a seguir a B si A y B sufren un cambio brusco en su trayectoria, la mejora propuesta para sortear este problema consiste en agregar un filtro, para diferenciar A y B por su color haciendo así más difícil que pierda a A. Para la implementación de este filtro de color comienza en el momento que el usuario selecciona el área a trackear, se calcula el color promedio de la zona elegida, y ese color es el que se le dará a la máscara, tal que sea lo único que sobreviva en el frame, esto además de aislar el objeto a trackear, si es un color distintivo tendrá una gran sinergia con el método de obtención de features de Shi-Tomasi dado que este frame tendrá bordes bien marcados. Volviendo al caso de los objetos A y B, si A y B son de colores distintos, con el filtro de color el tracker nunca detectará que existe un objeto B por lo que seguirlo será imposible, y por consecuencia seguirá únicamente a A, este comportamiento se puede ver en las

imagenes (8a) y (8b) donde se puede observar en la pantalla de debug como el único objeto con bordes definidos es el violeta.

Por otro lado, se obtuvo un algoritmo basado en los elementos mencionados previamente que permite detectar un objeto a partir de su color. El programa logrado permite al usuario seleccionar un área rectangular obtenida de una cámara o un video, siendo los elementos dentro de dicha sección los cuales son seguidos. De esta forma se pueden generar dos ventanas: donde simplemente se muestra el seguimiento del objeto en cuestión y otra resultante de lo denominado “modo de debug”, en la cual se observa el proceso de optical flow.



(a) Modo normal.



(b) Modo debug.

Figura 8: Resultado de la selección de un elemento.

De este modo, se observa en la Figura (8a) como se denota un objeto a seguir, mientras que la Figura (8b) muestra dos datos importantes. Por un lado, se observa en el mismo color del objeto seleccionado como se lo visualiza mediante optical flow, mientras que por otro lado, se muestran los puntos elegidos por el algoritmo de Shi-Tomasi en forma de puntos verdes, los cuales sirven para calcular el centro de masa a seguir. Cabe destacar que el filtro de Kalman estima la posición de este último. Adicionalmente el filtro de Kalman permite estimar la posición del objeto aunque este desaparezca de la pantalla al estar oculto de un objeto a partir de su velocidad estimada por el filtro y las ecuaciones de cinemática, contando con la habilidad de volver a encontrar al objeto cuando reaparezca.

El programa desarrollado permite elegir al usuario si desea o no ver la ventana en modo debug, la cual además informa sobre la posición del centro de masa. Para elegir un objeto es tan sencillo como seleccionar una área por sobre este, la cual aparece denotada en azul, para luego iniciar el seguimiento presionando al tecla “espacio”. Por otra parte, el programa permite reelegir un elemento a detectar presionando la tecla “r”.

- Descripción y análisis de lo original producido por el grupo.
- Simulaciones que justifiquen las ideas, y que prueben su originalidad.
- Análisis de resultados

#### IV. DESARROLLO

- Viabilidad, caminos alternativos.
- Proceso de implementación
- Documentación de los resultados: Resumen de lo más relevante, demos y programas van al apéndice.
- Evaluación y conclusiones del desarrollo.

##### IV-A. Posibles Mejoras

Algunas mejoras que pueden ser implementadas en futuros desarrollos son:

- Trackeado multiobjeto: Da la opción al usuario de trackear mas de un objeto en la pantalla.
- Filtro de Kalman donde la matriz de covarianza de medición no sea constante: En el modelo utilizado actualmente la matriz de covarianza es constante, esto asume que la precisión de medición es fija. Esto último no suele ser verdad en la realidad, una mejora es la introducción de modificaciones en la matriz de covarianza de manera dinámica así el filtro proporcionaría mejores predicciones, un caso en el que esto es vital es en la desaparición del objeto a trackear dado que en ese momento lo único utilizado es la predicción de Kalman.
- Filtro de sensibilidad Multicolor: Dado tres objetos  $A$ ,  $B$  y  $C$ , donde  $A$  cuenta con 2 colores distintivos, siendo estos rojo y azul,  $B$  un objeto rojo y  $C$  uno azul, lo que propone este filtro es la capacidad de diferenciar al objeto por mas de un color distintivo, cosa de que la interferencia de  $B$  o  $C$  en el camino de  $A$  no comprometa su seguimiento.
- Un mejor modelado de las ecuaciones dinámicas del filtro de Kalman para incluir la aceleración como variable de estado: El modelo utilizado actualmente consta de un movimiento rectilíneo uniforme bidimensional, la inclusión de la aceleración proporcionaría una mejor predicción.

##### IV-B. Conclusiones

Alan chupo pitino  $> 3 <$

Se logró seguimiento de un objeto especificado por el usuario en base a cualidades distintivas del mismo, siendo estas su color y su bordes, contando con una predicción certera en el caso de que se encuentre el objeto en pantalla al igual que la situación de que desaparezca de pantalla, aunque la varianza de la predicción aumentará cuanto mas tiempo no sea detectado el objeto, con la capacidad encontrar nuevamente el objeto.

La elección de los algoritmos y filtros utilizados son sinérgicos dado que estos se complementan entre sí, por ejemplo el filtro de color proporciona bordes definidos para el objeto a seguir, que es ideal para la obtención de bordes de Shi-Tomasi

## REFERENCIAS

- [1] R. C. Gonzalez, R. E. Woods y S. L. Eddins. *Digital Image Processing Using MATLAB*. Prentice Hall, 2da ed, 2002.
- [2] R. C. Gonzalez, R. E. Woods y S. L. Eddins. *Digital Image Processing Using MATLAB*. Prentice Hall, 2da ed, 2002, pp. 52-54.
- [3] D. H. Warren y Edward R. Strelow. *Electronic Spatial Sensing for the Blind: Contributions from Perception*. Springer Netherlands, 1er ed, 1985. pp. 414.
- [4] B. D. Lucas y T. Kanade. *An iterative image registration technique with an application to stereo vision*. De Proceedings of Imaging Understanding Workshop, pp. 121-130.
- [5] W.S.P. Fernando, L. Udawatta , P. Pathirana. *Identification of Moving Obstacles with Pyramidal Lucas Kanade Optical Flow and k means Clustering*. Faculty of Engineering, University of Moratuwa, Sri Lanka.
- [6] G. Terejanu, "Discrete Kalman Filter Tutorial", Cse.sc.edu, 2020. [Online]. Available: <https://cse.sc.edu/terejanu/files/tutorialKF.pdf>. [Accessed: 15- Jun- 2020].
- [7] G. Welch and G. Bishop, *An Introduction to the Kalman Filter*. University of North Carolina at Chapel Hill: Department of Computer Science, 2006.