12º Congreso Nacional de Mecatrónica, ISBN 978-607-95347-8-3
Octubre 10 y 11, 2013. León, Guanajuato.

Procesamiento y análisis de imágenes mediante el algoritmo SIFT en OpenCV.

Palma Olvera Raúl David, Delgado Rosas Manuel, Pedraza Ortega Jesús Carlos, Aceves Fernandez Marco Antonio, Tovar Arriaga Saúl

Universidad Autónoma de Querétaro, Facultad de Informática, Campus Juriquilla.

Resumen

En el presente trabajo de investigación se describe el desarrollo del algoritmo SIFT (Scale Invariant Feature Transformation), los diferentes procesos que lo conforman, así como una serie de pruebas realizadas con implementaciones de este mismo en OpenCV.

Se desarrollaran una serie de comparaciones de un objeto base o plantilla, y una imagen fija de una escena, esta con variaciones en el ángulo de la imagen, así como la posición del objeto.

Una vez realizadas las pruebas se comparara y analizara el nivel de error que puede presentarse en las pruebas, conforme a las variaciones de orientación, ángulo del objeto e iluminación

Palabras clave: OpenCV, Visión por Computadora, SIFT, Procesamiento de Imágenes, *Scale Invariant Feature Transformation.*

1. Introducción

En la actualidad ha incrementado el uso de métodos para el reconocimiento de un objeto en una escena, uno de ellos es el algoritmo SIFT propuesto por Lowe [1], este ha logrado establecerse como un estándar para dicho propósito debido a su alta precisión y su bajo tiempo de procesamiento.

Así mismo el uso de algoritmos de reconocimiento de puntos de interés en un objeto y se ha extendido a los dispositivos móviles, debido a su capacidad de procesamiento y su cámara integrada.

Existen varios trabajos relacionados con el uso de móviles y el algoritmo SIFT, como el trabajo de Takacs [2], el cual implementa en un móvil un algoritmo SIFT el cual hace una comparación con ciertas imágenes contenidas en una base de datos arrojando como resultado el nombre del lugar en donde se encuentra, obteniendo resultados satisfactorios, en las condiciones optimas.

Muchas metodologías y mejoras al algoritmo SIFT se han desarrollado, como en el trabajo de Zhang [3], el cual uso un enfoque basado en un "histograma de localizador de color" que era usado para limitar la búsqueda en la base de imágenes, con un paso final basado en el algoritmo SIFT, otro caso es el de He [4], también utilizo el algoritmo SIFT pero empleo un método de aprendizaje en el tiempo para encontrar "características prototipo" las cuales fueran utilizadas para la localización y solucionar las variaciones que se presentaban conforme a los cambios naturales presentados en el lugar. Ambos obteniendo resultados satisfactorios al enfrentar los cambios de iluminación en el exterior.

2. Algoritmo SIFT

Como hace mención en su investigación Herigert [5], para lograr detectar una imagen, en primer lugar es necesario encontrar los puntos clave o de interés que identifiquen de una manera unívoca a cada uno de los objetos de manera de poder encontrarlos nuevamente si estos aparecen en cualquier otra escena.
La idea principal del algoritmo SIFT es la transformación de la imagen a una presentación compuesta de puntos de interés, estos puntos contienen la información característica de la imagen que luego son usados para la detección de muestras Jinxia [6]. El algoritmo se realiza mediante 4 pasos principalmente:

Figura 1: Diagrama de la metodología del algoritmo SIFT.

2.1 Construcción de Pirámides de Scale-Space y Detección de los Extremos Locales.

Se representa la imagen en diferentes escalas y tamaños. Se lleva a cabo de manera eficiente mediante el uso de la función de diferencia Gaussiana, para identificar los posibles puntos de interés que son invariables a escala y orientación.

Para la detección de extremos locales se debe trabajar con la imagen original filtrada. El único filtro apropiado para estos efectos son los filtros Gaussianos de low pass. Se utiliza este tipo de filtros debido a que la función Gaussiana es invariante a escala en el espacio (Figura 2), para la detección de puntos de interés. Además, elimina el ruido de la imagen.

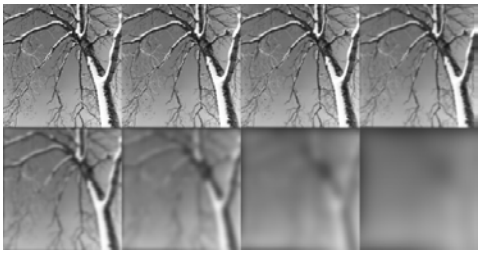

Figura 2: Imagen en diferentes escalas.

La imagen con la que se trabaja es la convolución entre la imagen original y el filtro Gaussiano.

Adicionalmente, se utiliza una distinta desviación estándar para el filtrado de la diferencia de dos filtros Gaussianos.

Otra manera de trabajar estas imágenes es utilizando el Laplaciano de funciones Gaussianas, sin embargo, la utilización de esto es lenta y la diferencia de Gaussianas es una aproximación bastante eficiente. Además de la similitud podemos ver, qué impacto en la imagen tiene la convolución con este filtro. La transformada de Fourier de las funciones tiene la misma forma (Figura 3), porque son funciones Gaussianas Fernández [7].

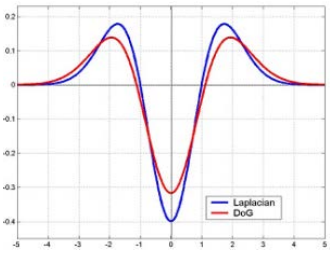

Figura 3: Diferencia entre la función Laplaciana y la diferencia Gaussiana.

En una imagen real, las frecuencias que contienen la información característica son las medias. El ruido digital está en las frecuencias altas, que está en cada imagen digital. Por otro lado, las frecuencias bajas, contienen variaciones suaves por lo que no son relevantes para la obtención de los puntos de interés.

La convolución con la DoG (Diferencia Gaussiana) se hace para toda la imagen y con escalas diferentes, para detectar estructuras en todos los lugares y con todos los tamaños. Para una implementación rápida, el algoritmo trabaja cada octava en forma individual, como vemos en la Figura 3.

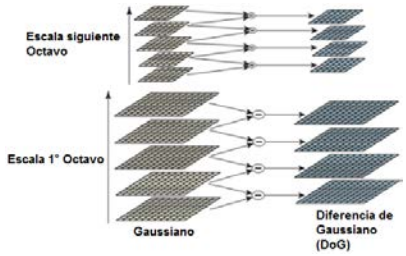

Figura 4: Por cada Octava en diferentes escalas se aplica una Diferencia Gaussiana (DoG).

Como menciona Lowe [1] [8], en sus investigaciones, para cada octava un número de 3 escalas y un sigma de 1.6 eran los valores óptimos. Por eso, primero se hace la convolución con 3 escalas de las Gaussianas, y luego para obtener la DoG se hace con la resta de imágenes vecinas como se puede observar en la Figura 5 donde se ve el resultado que se obtiene tratando imágenes reales.

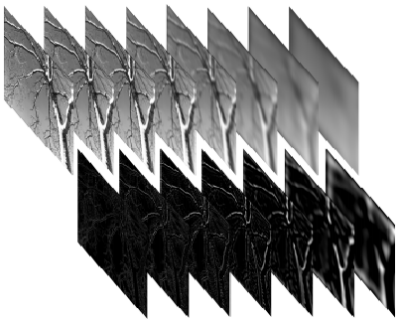

Figura 5: Como se ve reflejado la Diferencia Gaussiana en imágenes reales.

Para la octava siguiente, solamente hay un nuevo muestreo con un factor 2 y la repetición de la resta. Con este método se crean muchas imágenes filtradas con valores extremos donde el tamaño y el lugar de la DoG es similar a la estructura dentro de la imagen.

Para buscar los extremos en las imágenes convolucionadas, cada píxel es comparado con todos sus píxeles vecinos, ambos en el dominio del espacio y en del dominio de la escala (Figura 6). Solo si todos tienen un valor distinto, este lugar va a pasar el examen.

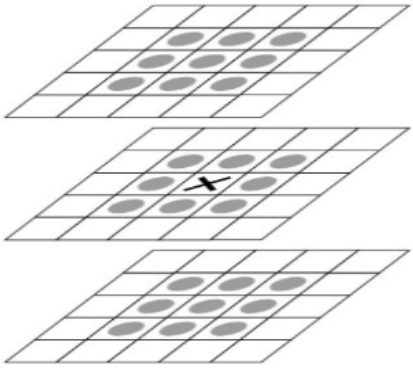

Figura 6: Cada pixel de la imagen es comparado con su pixel vecino en la Pirámide de DoG.

2.2 Localización de Puntos Clave.

Entre los puntos que sobrevivieron el examen de la búsqueda de extremos hay muchos que caracterizan puntos con poco contraste. Estos no son estables, si la iluminación cambia producen ruido.

Para quitarlos se examina primero si el máximo o mínimo está en un lugar entre esos píxeles, para estimar la función D con una serie de Taylor de grado 2.

$$D(x) = D + \frac{\partial D^T}{\partial x}x + \frac{1}{2}x^T\frac{\partial^2 D}{\partial x^2}x \quad (1)$$

Después de la derivación de esta aproximación e igualando a cero queda:

$$\hat{x} = -\frac{\partial^2 D^{-1}}{\partial x^2} * \frac{\partial D}{\partial x} \quad (2)$$

$$\rightarrow D(\hat{x}) = D + \frac{1}{2} * \frac{\partial D^T}{\partial x}\hat{x} \quad (3)$$

Si el valor de D(x^) es menor a 0.03, el punto es eliminado, suponiendo que D tiene valores de 0 a 1.

Además de quitar aquellos puntos con poco contraste, hay que encontrar y descartar candidatos que vienen de una línea recta y no de una esquina. Si hay una línea recta, la curvatura de D va a ser grande en una dirección pero pequeña en la que es perpendicular.

2.3 Asignación de Orientación.

En este paso se asigna una dirección a cada punto de interés el cual depende de las muestras de los puntos que se poseen en su entorno.

Para tener un buen descriptor, la localización y la función local de aproximación que tenemos hasta ahora no son suficientes. Hay también que examinar el valor del gradiente y su orientación. El valor corresponde a la escala de la Gaussiana y mediante ese tratamiento la descripción del punto de interés es invariante con respecto a la escala. Además con el conocimiento de la orientación, la caracterización es independiente con respecto a la dirección.

Gracias al uso de una ventana Gaussiana, se le asignan valores a los pixeles entre más lejanos tienen un impacto más pequeño que los píxeles cercanos. Con un histograma de orientación con ventanas de 10 grados el algoritmo trata de buscar la dirección verdadera usando una interpolación de los 3 valores más grandes del histograma.

El histograma de orientación es formado por la orientación del gradiente de los puntos muestreados en la región del punto clave. Cada pico en el histograma corresponde a la dirección dominante del gradiente local (un pico con el 80% es usado para crear el punto clave con su orientación). Sólo alrededor del 15% de los puntos se les asignan múltiples orientaciones, en la Figura 7 se aprecia una representación de los puntos clave y su orientación, donde no importa la rotación del a imagen, los puntos siguen siendo identificados correctamente Yu [9].

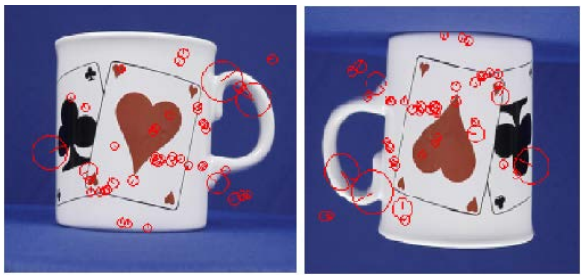

Figura 7: Representación de puntos clave con sus respectivas direcciones.

2.4 Descripción de Puntos Clave

Para tener una descripción que sea lo adecuadamente invariante a cambios en un punto de vista en el espacio, un tratamiento relacionado a la función de las neuronas en la visión biológica es usado.

Debido a que es necesaria la independencia de pequeñas translaciones del punto de interés, Edelman[10] propuso el algoritmo siguiente.

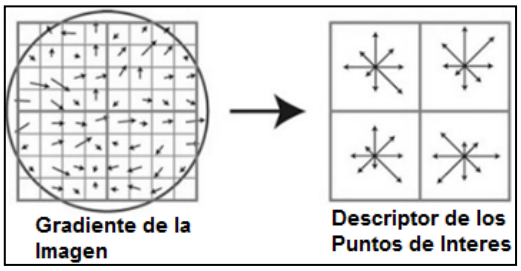

Figura 8: Usando una ventana Gaussiana, los valores m y Θ se examinan en la vecindad del punto de interés.

Una ventana Gaussiana representada por el círculo en la Figura 8, selecciona los valores de m y Θ prefiriendo los que están más cerca del centro. Después sigue una distribución en sectores más grandes y otra vez el uso de un histograma con 8 distintas orientaciones.

La ventaja de esto es que los histogramas quedan iguales, aún cuando el centro de la ventana Gaussiana se mueve hasta 4 píxeles. Esto hace la descripción bastante robusta con respecto a las translaciones por cambios de puntos de vista. La Figura 8 muestra un ejemplo reducido a 2x2 histogramas. Según el algoritmo SIFT se tratará con 4x4 con 8 posibles direcciones correspondiente a un vector de 4x4x8 = 128 dimensiones para cada punto de interés.

3. Análisis de resultados

Se realizaron las pruebas correspondientes en el software de OpenCV, se compararon una serie de objetos en diferentes imágenes con una implementación de SIFT con los valores por default obteniendo los siguientes resultados:

Objetos	Total	Correctos	Porcentaje (%)
Vanellope	626	259	41.37%
Rana	926	686	74.08%
Bote de Café	573	172	30.01%

Tabla 1: Valores totales y correctos encontrados entre los diferentes objetos utilizados en las pruebas.

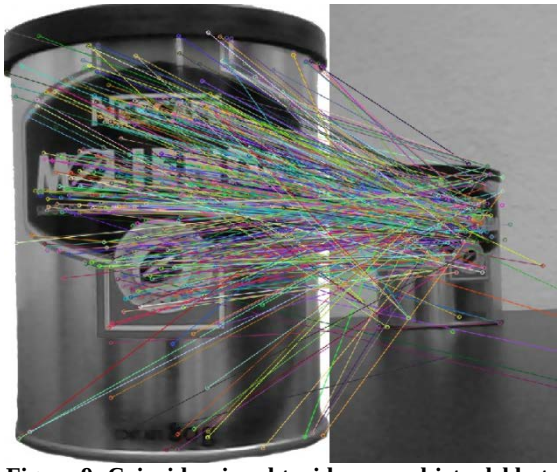

Figura 9: Coincidencias obtenidas con objeto del bote de café utilizando los valores por defecto de SIFT.

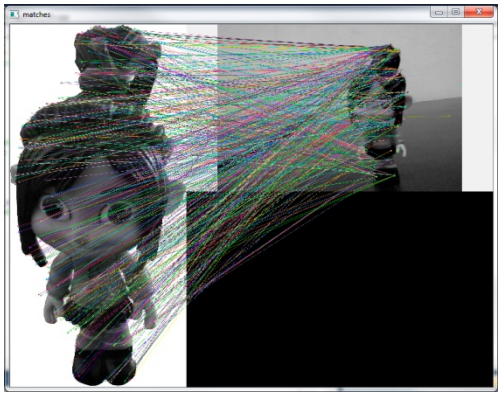

Figura 10: Emparejamiento obtenido utilizando una figura de juguete (Vanellope).

Posteriormente se realizaron las mismas pruebas modificando en cada caso los valores por default para obtener una mejor identificación del os puntos de interés importantes del objeto, obteniendo los resultados siguientes:

Objetos	Total	Correctos	Porcentaje (%)
Vanellope	309	111	35.92%
Rana	318	143	44.96%
Bote de Café	183	44	24.04%

Tabla 2: Cantidad de puntos encontrados entre los objetos con diferentes valores de SIFT.

Al realizar las pruebas con valores modificados la cantidad de puntos de interés se vio notablemente reducida, al igual que las coincidencias, obteniendo valores más bajos, esto se debe a que al igual que se eliminaron varios puntos que estaban dando resultados falsos correctos.

Figura 11: Puntos de interés más importantes resaltados con valores diferentes a los por defecto.

SIFT_contrastThreshold	0.11
SIFT_edgeThreshold	7.00
SIFT_nOctaveLayers	2
SIFT_nfeatures	0
SIFT_sigma	1.60

Figura 12: Parámetros utilizados en la comparación de las imágenes del bote de café.

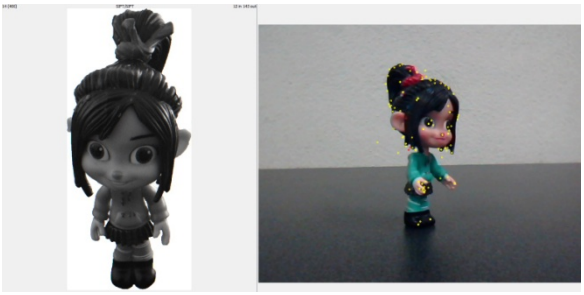

Figura 13: Puntos de interés obtenidos en la comparación en una posición diferente a la foto original.

SIFT_contrastThreshold	0.10
SIFT_edgeThreshold	9.00
SIFT_nOctaveLayers	4
SIFT_nfeatures	0
SIFT_sigma	1.60

Figura 14: Parámetros utilizados en la comparación de la figura de juguete.

Así mismo se realizaron una serie de pruebas con la imagen de la figura de juguete rotada en 5 hasta 180 grados, donde se comprueba que la implementación del algoritmo SIFT analizada, tiene un error aproximado del 5 al 10 por ciento, haciéndolo casi invariable a la rotación.

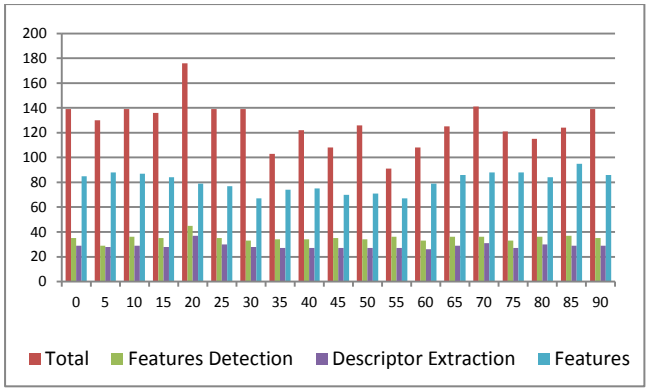

Tabla 3: Rango de características de la imagen rotada de 0 a 90 grados.

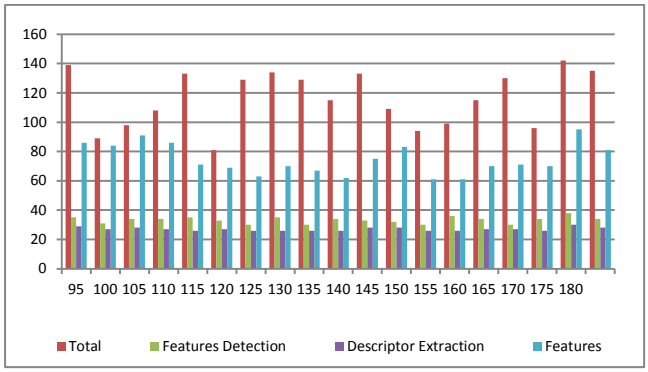

Tabla 4: Rango de Valores de la imagen rotada de 95 a 180 grados.

4. Conclusiones

El propósito al hacer las comparaciones en esta investigación del algoritmo SIFT, fue para identificar diferentes objetos mediante sus puntos de interés en una serie de imágenes con variaciones en la rotación y posición del objeto, hacer un análisis de cómo altera en el reconocimiento de las características el modificar los valores por defecto de dicho algoritmo, esto mediante el software de OpenCV y Visual Studio 2010,

Los resultados obtenidos nos muestran que el algoritmo SIFT es prácticamente invariable a los cambios de rotación en la imagen a comparar, así mismo en la rotación y escala del objeto analizado, ya que en las pruebas solo obtuvimos de 5 a 10 por ciento como rango de error, esto aporto una base que nos servirá para en un futuro, una mejora a dicho algoritmo en cuestión de la fiabilidad en los resultados correctos verdaderos, así mismo una mejor respuesta a los cambios en la iluminación y orientación de los objetos.

Referencias

[1] Lowe, D. *"Object Recognition from Local Scale-Invariant Features"*, Computer vision. The Proceedings of the Seventh IEEE International Conference on (Vol. 2, pp. 1150-1157), IEEE. Computer Science Department, University of British Columbia, Vancouver, B.C., Canada. 1999.

[2] Takacs, G. Xiong, Y. Grzeszczuk, R. Chandrasekhar, V. Chen, W. Pulli, K. Gelfand, N. Bismpigiannis, T and Girod, B. *"Outdoors Augmented Reality on Mobile Phone using Loxel-Based Visual Feature Organization"*, Proceedings of the 1st ACM International Conference on Multimedia Information Retrieval, ACM. (pp. 427-434). California, Estados Unidos, 2010.

[3] Zhang, W and Kosecká, J. *"Localization Based on Building Recognition"*, Computer Vision and Pattern Recognition-Workshops CVPR Workshops, IEEE Computer Society Conference, IEEE. (pp. 21-21). Department of Computer Science, George Mason University, Fairfax en Estados Unidos, 2005.

[4] He, X., Zamel, R. and Mnih, V. *"Topological Map Learning from Outdoor Image Sequences"*, Journal of Field Robotics, 23(11-12), 1091-1104. University of Toronto, Toronto, Ontario, Canada, 2006.

[5] Herigert, M. Bouchet, N. y Pianetti, I. *"Reconocimiento de Imágenes mediante Scale Invariant Feature Transformation (SIFT)"*. Unpublished Manuscript, preprint at http://www.frsf.utn.edu.ar/Fcneisi2010/Farchivos/F04-Reconocimiento_de_Imgenes_SIFT.pdf UTN Facultad Regional Concepción del Uruguay, 2010.

[6] Jinxia, L and Yuehong, Q. *"Application of SIFT Feature Extraction Algorithm on the Image Registration"*. IElectronic Measurement & Instruments (ICEMI), 2011 10th International Conference on (Vol. 3, pp. 177-180). IEEE. Xi'an Institute of Optics and Precision Mechanics of CAS, China, 2011.

[7] Fernández, A., González, P., Ruiz, V. y Ortega, J. *"Iris Recognition Based on SIFT Features"*. Biometrics, Identity and Security (BIdS), 2009 International Conference on (pp. 1-8). IEEE. Escuela Politécnica Superior Universidad Autónoma de Madrid, España, 2009.

[8] Lowe, D. *"Distinctive Image Features from Scale-Invariant Keypoints"*. International Journal of Computer Vision, 60(2), 91-110. Computer Science Department, University of British Columbia, Vancouver, B.C., Canada, 2004.

[9] Yu, G and Morel, J. *"A Fully Affine Invariant Image Comparison Method"*. Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on (pp. 1597-1600). IEEE. CMAP, Ecole Polytechnique, Francia, 2010.

[10] Edelman, S., Intrator, N. and Poggio, T. *"Complex cells and object recognition"*. Unpublished Manuscript, preprint at http://www.ai.mit.edu/edelman/mirror/nips97.ps.Z Massachusetts Institute of Technology, Cambridge, Massachusetts, Estados Unidos, 1997.