

Detección de Productos en Supermercados

Juan Pablo Vasquez Fuentes and Ralf C. Wilke

Pontificia Universidad Católica de Chile, Departamento de Ingeniería Eléctrica,
Santiago de Chile, Chile
2008

jpvasquezf@gmail.com, ralf@ralfwilke.com
<http://www.ralfwilke.com/chile/puc>

Abstract. Reducing the time of permanency in supermarkets has been investigated for a long time. Using the results of stochastic models or introducing complex electric systems like attaching RFID transponders to every product has helped to solve this problem. Instead of the manual barcode detection, the transponders identify themselves to the system transmitting a unique serial number. A known problem of this method is the changing of radio wave propagation, which occurs with different scenarios of humans and shopping carts.

In this context, your work offers a new alternative for product recognition. Instead of radio waves, we use a picture taken by a CCD camera, which is mounted above the shopping cart. The image data is sent via USB cable to a personal computer, which does all the image processing. First, we created a database containing the keypoints for object recognition using the SIFT algorithm published in 2003. In this work, an amount of 10 products was examined, but the number can be extended easily. The image taken by the camera is first passed through a preprocessing step, modifying the contrast and correcting the gamma value of the camera. This helps to become mostly independent of changing illumination or contrast. SIFT algorithm is used to extract keypoints of the image containing the products. Next step is the matching of the keypoints with the contents of the database. Assuming that every product has a distinctive package, automated product recognition can be done. If the quantity of keypoints is above a certain threshold, we can conclude that the product is present in the considered image. Finally, the total value of the sale is calculated by a lookup table containing the price of every product. Using modern methods like GPU computing for SIFT, the total time of permanency in supermarkets can be reduced seriously.

Key words: SIFT, Reconocimiento de objetos, Sistemas de cobranza

1 Introducción

Minimizar el tiempo de permanencia en los supermercados es un tema que se ha investigado por muchos años. Desde estudios de modelos estocásticos hasta complejos sistemas eléctricos has contribuido a este problema.

El “Supermercado Inteligente” fue implementado en Alemania, donde la compra se realiza en un tiempo mucho menor al de los supermercados tradicionales básicamente debido a que los compradores no tienen que pasar por las cajas, pasar todos los productos y pagar por ellos, los productos son identificados por medio de RFID (identificación por medio de frecuencia de radio), entonces, al momento de abandonar el lugar, serán cobrados los productos detectados. RFID no sólo es utilizado en los ya mencionados “Supermercados Inteligentes”. Además es muy útil en el manejo de stock en grandes bodegas. Este sistema es muy efectivo, pero esta tecnología no ha llegado al país. Cada producto debe tener un pequeño emisor de señal el cual es captado por un portal, el que detecta las señales emitidas. Este sistema es de muy bajo costo, ya que los emisores y receptores tan sólo cuestan centavos de dólar, pero uno de sus problemas es que hay materiales que las ondas emitidas no pueden atravesar.

En este contexto, nuestro proyecto consiste en buscar una alternativa eficiente a este método reemplazando el reconocimiento de los productos por medio de radio frecuencia y reconocerlos por medio de imágenes. Se reemplazarían los emisores de ondas de radio y el portal receptor simplemente por una pequeña cámara.

El objetivo principal es detectar, por medio de una Web-Cam alrededor de 10 productos distintos y conociendo el precio individual de cada producto, calcular el valor total de la compra. Asumiendo que cada producto tiene un paquete o envoltorio distinto, debe ser posible reconocerlo por medio de imágenes. Uno de nuestros objetivos es reducir el tiempo de permanencia de los usuarios en los supermercados, por lo que es muy importante que la detección se haga de la forma más rápida posible.

El procedimiento consistirá en obtener imágenes de los productos en el interior de un carro. De esta manera, mediante un procedimiento de reconocimiento de patrones, se pretende identificar la presencia de cada producto y en qué cantidad se encuentra. El reconocimiento de los distintos productos se realizará mediante el método SIFT (Scale-invariant feature transform).

Con este proyecto buscamos lograr una eficiente detección de distintos productos y así calcular el valor total de la compra sin la necesidad de sacar los productos del carro. Para que el proyecto sea exitoso y aplicable en la realidad debe tener un rendimiento del 100%.

2 Estado del arte

Para lograr detectar objetos en una imagen, en primer lugar es necesario encontrar puntos de interés. Metodologías para lograrlo se han ido desarrollando desde el año 1981 con el trabajo de Moravec [11], quien fue el primero en desarrollar algoritmos de búsqueda de vértices en imágenes, con el fin de hacer coincidir imágenes obtenidas desde distintos ángulos. El trabajo de Moravec fue años más tarde mejorado por Harris y Stephens (1988) [12], creando el detector de esquinas de Harris, muy utilizado para estos ámbitos, que no sólo busca esquinas como dice su nombre, además busca cualquier punto dentro de la imagen

con altos gradientes en todas direcciones. Pero para detectar los puntos de interés de una imagen, líneas rectas no sirven porque los puntos en ellas son muy parecidos a sus vecinos. Entonces ellas no sirven para una buena descripción de las características que contiene una imagen. Además, el detector de Harris no es invariante para cambios de escala, que pueden ocurrir en una cantidad muy grande de imágenes reales tomadas con una cámara desde distancias diferentes.

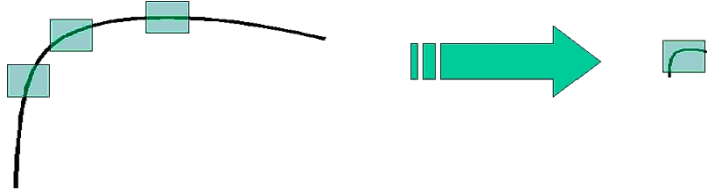


Fig. 1. Con el detector de Harris, curvas con escalas diferentes pueden ser detectadas como una esquina, pero también como un borde, si el tamaño es grande. Fuente: [4]

2.1 Scale Invariant Feature Transform (SIFT)

Para mejorar este problema, Lowe [1] encontró en 2003 un nuevo algoritmo denominado SIFT, que revolucionó el mundo del procesamiento de imágenes. Su idea principal es la transformación de la imagen a una representación compuesta de "puntos de interés". Esos puntos contienen la información característica de la imagen y luego son usados para la detección de muestras. El algoritmo se realiza mediante 4 pasos:

- Pre-procesamiento y detección de los extremos locales
- Entre ellos, localización de los puntos de interés, que describen bien las características de la imagen.
- Asignación de orientación de los gradientes en la vecindad de estos puntos.
- Descripción de las características de una imagen en un vector.

Los pasos están explicados con más detalle en el parte siguiente.

Detección de los extremos locales Para la detección de extremos locales se debe trabajar con la imagen original filtrada. El trabajo de Koenderink (1984) [13] y Lindeberg (1994) [14] encontró que el único filtro apropiado para estos efectos son los filtros Gaussianos pasabajos. Se utiliza este tipo de filtros debido a que la función Gaussiana es invariante a escala en el espacio, para la detección de puntos de interés. Además, elimina el ruido de la imagen. La imagen con la que se trabaja es la convolución entre la imagen original y el filtro Gaussiano. Si llamamos la imagen original $I(x, y)$ y al filtro $G(x, y, \sigma)$, entonces la imagen con la que se trabajará $L(x, y, \sigma)$ resulta de la convolución entre ambas, es decir:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \quad (1)$$

$$G(x, y, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} \cdot \left(-\frac{x^2 + y^2}{2\sigma^2} \right) \quad (2)$$

Adicionalmente, el trabajo de Lowe propuso la utilización para el filtrado la diferencia de dos filtros Gaussianos, con distinta desviación estándar cuya diferencia es un factor k.

$$D(x, y, \sigma) = G(x, y, k\sigma) - G(x, y, \sigma) \quad (3)$$

$$L(x, y, \sigma) = D(x, y, \sigma) * I(x, y) \quad (4)$$

Otra manera de trabajar estas imágenes es utilizando el Laplaciano de funciones Gaussianas, sin embargo la utilización de éste es bastante lento y la diferencia de Gaussianas es una aproximación bastante eficiente para estos fines. El Laplaciano tiene la formula siguiente con G como definida antes.

$$M = \sigma^2 \cdot \Delta G = \sigma^2 \left(\frac{\partial^2 G}{\partial x^2} + \frac{\partial^2 G}{\partial y^2} \right) \quad (5)$$

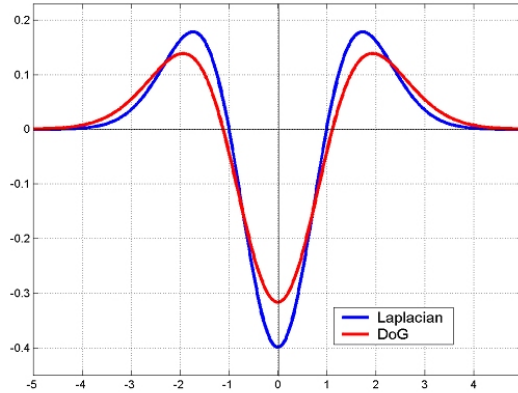


Fig. 2. La Laplaciana M es muy parecido a la diferencia de gaussianas (DOG), definido como D antes. Fuente: [4]

En figura 2 mostramos las dos alternativas. 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, porque son funciones Gaussianas. Entonces, podemos interpretar la figura 2 también como la respuesta al impulso en del dominio del espectro. 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 culpa de la cuantificación, no va a pasar el filtro. Por otro lado, las frecuencias bajas, contienen variaciones suaves por lo que no son relevantes para la obtención de los puntos de interés.

Como se puede ver fácilmente, la aproximación de las dos fórmulas sigue de los cálculos siguientes usando la definición de la derivada:

$$\sigma \cdot \Delta G = \frac{\partial G}{\partial \sigma} \approx \frac{G(x, y, k\sigma) - G(x, y, \sigma)}{\sigma(k - 1)} \tag{6}$$

$$\implies G(x, y, k\sigma) - G(x, y, \sigma) \approx (k - 1) \cdot \sigma^2 \cdot \Delta G \tag{7}$$

La convolución con la DOG 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. Investigaciones de Lowe dieron que 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. 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 el dominio de la escala. Solo si todos tienen un valor distinto, este lugar va a pasar el examen. Figura 3 explica este procedimiento.

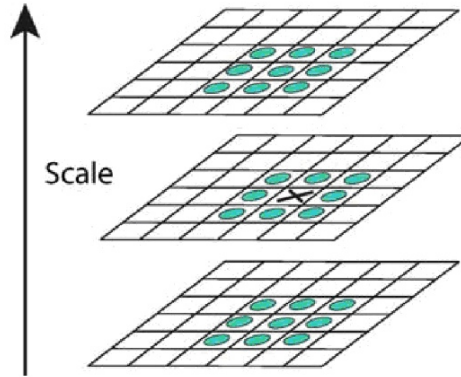


Fig. 3. Buscar por extremos en todos los lugares vecinos

Detección de puntos de interés Entre los puntos que sobrevivieron el examen de buscar extremos hay muchos que caracterizan puntos con poco contraste. Ellos no son estables si por ejemplo la iluminación cambia muy poco y produce ruido. Para sacarlos se examina primero si el máximo o mínimo está en un lugar entre píxeles, para estimar la función D con una serie de Taylor de grado 2.

$$D(x) = D + \frac{\partial D^T}{\partial x} x + \frac{1}{2} x^T \frac{\partial^2 D}{\partial x^2} x \quad (8)$$

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

$$\hat{x} = -\frac{\partial^2 D^{-1}}{\partial x^2} \cdot \frac{\partial D}{\partial x} \quad (9)$$

$$\implies D(\hat{x}) = D + \frac{1}{2} \cdot \frac{\partial D^T}{\partial x} \hat{x} \quad (10)$$

Si el valor de $D(\hat{x})$ es menos de 0.03, el punto es eliminado, suponiendo que D tiene valores de 0 a 1.

Además de sacar 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 pequeño en la que es perpendicular. Este factor corresponde a un valor propio α grande y un β más pequeño de la matriz Hessiana. Con el conocimiento de:

$$Trace(H) = \frac{\partial^2 D}{\partial x^2} + \frac{\partial^2 D}{\partial y^2} = \alpha + \beta \quad (11)$$

$$Det(H) = \frac{\partial^2 D}{\partial x^2} \cdot \frac{\partial^2 D}{\partial y^2} = \alpha \cdot \beta, \alpha = r\beta \quad (12)$$

Para un r predefinido hay que examinar si se cumple la inecuación dada por:

$$\frac{Trace(H)^2}{Det(H)} < \frac{(r+1)^2}{r} \quad (13)$$

Lowe propone en [1] un umbral de $r = 10$.

Asignación de orientación Para tener un buen descriptor de un punto de interés, el lugar 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 con 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. Según las leyes de la geometría en un espacio vectorial de Euclides, se calcula el valor $m(x, y)$ y la orientación $\Theta(x, y)$ de una función discreta como:

$$m(x, y) = \sqrt{(\Delta L_x)^2 + (\Delta L_y)^2} \quad (14)$$

$$= \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2} \quad (15)$$

$$\Theta(x, y) = \tan^{-1} \left(\frac{\Delta L_y}{\Delta L_x} \right) \quad (16)$$

$$= \tan^{-1} \left(\frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)} \right) \quad (17)$$

Para tener un conocimiento más general de la orientación en un punto de interés, el algoritmo SIFT examina también los valores m y Θ de los píxeles que están en una región al lado. Gracias a una ventana gaussiana usada, los valores de píxeles más lejanos tienen un impacto más pequeño que píxeles cercanos. Con un histograma de la orientación con cajas de 10 grados el algoritmo trata de buscar la dirección verdadera usando una interpolación de los 3 valores más grandes de el histograma. Con éste, también direcciones pueden ser detectadas que están entre dos cajas de el histograma. En los experimentos de Lowe este procedimiento resulta bien.

Descripción de puntos de interés: Para tener una descripción que será bastante invariante a cambios en el punto de vista en el espacio, un tratamiento relacionado a la función de las neuronas en la visión biológica es usada. Debido a que es necesaria la independencia de pequeños translaciones del punto de interés, Edelman et al. [15] propusieron en 1997 el algoritmo siguiente.

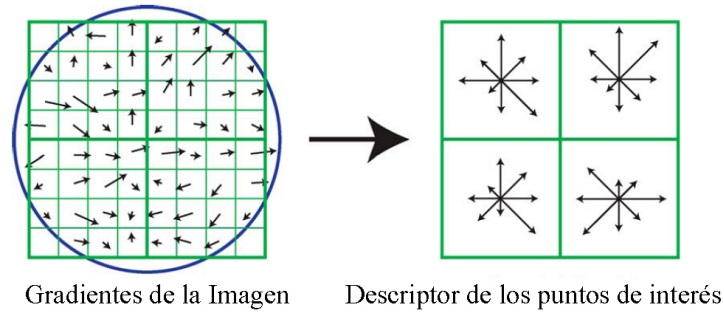


Fig. 4. Usando una ventana gaussiana, los valores m y Θ examina en la vecindad del punto de interés. Después sigue un tratamiento con histogramas para 8 orientaciones distintas [1].

Una ventana gaussiana representada por el círculo en la figura 4, 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. Este hace la descripción bastante robusta con respecto a las translaciones por culpa de cambio del punto de vista. La figura 4 muestra un ejemplo reducido a 2x2 histogramas. Según el algoritmo SIFT vamos a tratar con 4x4 con 8 posibles direcciones correspondiente a un vector de $4 \times 4 \times 8 = 128$ dimensiones para cada punto de interés.

Aplicación al reconocimiento de objetos Para examinar si una imagen contiene un patrón hay que hacer tres pasos de trabajo. Primero hay que sacar las características de las imágenes con las muestras y guardarlas en una base de datos. Esta base está fija e inmutable. Para la detección en una imagen hay que sacar también los puntos de interés y describirlos con el algoritmo SIFT. El desafío es de buscar en la base de datos, si hay correspondencias. El espacio de los descriptores tiene 128 dimensiones entonces es muy difícil de buscar por correspondencias de manera efectiva. Primero se busca los descriptores vecinos que están lo más cerca. Para eliminar errores, se examina también la distancia hasta el vecino segundo más cerca. Si la proporción de las distancias es casi 1, hay un error. El valor es normal entre 0.3 y 0.5. Para saber después si la coherencia está limitada a un área finita, se usa la transformada de Hough, resultando un conglomerado con al menos 3 entradas. Todavía es necesario examinar si la suposición de haber encontrado la muestra correcta. Propuesto por Lowe en 1999 [2] se trata de reproducir y verificar la distorsión afín hecho por un punto de vista diferente. El modelo de la transformada afín es:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} m_1 & m_2 \\ m_3 & m_4 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \quad (18)$$

donde $[x \ y]^T$ es el punto original, la matriz \mathbf{M} contiene toda la distorsión y $[t_x \ t_y]^T$ representa una translación resultando en un punto de imagen $[u \ v]^T$. Como sabemos algunos candidatos para $[x \ y]^T$ y $[u \ v]^T$, podemos tratar de examinar los parámetros de la transformación según:

$$\begin{bmatrix} x & y & 0 & 0 & 1 & 0 \\ 0 & 0 & x & y & 0 & 1 \\ & & \dots & & & \end{bmatrix} \cdot \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ t_x \\ t_y \end{bmatrix} = \begin{bmatrix} u \\ v \\ \vdots \end{bmatrix} \implies Ax = B \quad (19)$$

Tratando como un sistema lineal la solución para x con el error cuadrado minimizado es:

$$x = [A^T A]^{-1} A^T b \quad (20)$$

Comparando el error con el valor medio del límite usando la Transformada de Hough, equivocaciones pueden ser detectadas y eliminadas. Finalmente viene una

evaluación de la probabilidad según un modelo de Lowe de 2001 [2]. Solamente si el valor es más de 98 %, vamos a aceptar el reconocimiento de la muestra en la imagen para examinar.

2.2 Aplicaciones reconocimiento de patrones

Junto con del desarrollo de métodos del reconocimiento de características en imágenes, había una gran cantidad de trabajo en sus aplicaciones. Por ejemplo, la patente EP0758081 [5] se refiere a la identificación de objetos con un análisis del espectro reflejado por el objeto. Para la iluminación, una fuente de luz especial emitiendo un espectro amplio es necesario. La imagen capturada de una cámara CCD es procesado después para la detección. La necesidad de una fuente de luz especial da una gran desventaja de este método. Nosotros vamos a usar una lámpara normal con una ampolleta. Después de la publicación de SIFT, muchas aplicaciones aparecieron que funcionan basándose en este algoritmo. Hay por ejemplo un método de detección y de reconocimiento de fachadas de casas [6] o la detección de huellas digitales [7], que se usan para sistemas de control de acceso a lugares especiales. En vez de usar los dedos de una persona para una característica biométrica, también se usan el iris para la identificación [9]. En los últimos meses, una gran cantidad de trabajo se ha estado haciendo con respecto a la computación de SIFT, no por medio de la CPU de un computador, en vez se utiliza una GPU (Graphic Processing Unit) de una tarjeta gráfica [3], [6]. Para el algoritmo SIFT, muchos cálculos pueden ser hechos al mismo tiempo. La arquitectura de una GPU está diseñado especialmente para este procesamiento paralelo, que reduce mucho el tiempo necesitado para los cálculos. Nosotros no vamos a usar esta ventaja, porque no poseemos un computador con una buena tarjeta gráfica.

3 Método propuesto

3.1 Características base de los Productos

Para poder identificar los productos presentes en la imagen captada, es necesario encontrar los puntos de interés de cada producto. Para ello se realiza un procedimiento a cada una de las imágenes de referencia, a través del método SIFT. Para cada producto, más de 1500 puntos de interés fueron encontrados. Este procedimiento era bastante lento, sin embargo no era necesario realizarlo cada vez que se ejecutaba el programa, ya que el resultado de la búsqueda era almacenado pasando a formar parte de la base de datos necesaria para la detección de productos. Al no tener que realizar la búsqueda de dichos puntos, se reduce bastante el tiempo de ejecución del programa.

3.2 Preprocesamiento de la imagen

Para ser independiente a cambios en la iluminación y para mejorar la imagen capturada de la web-cam, sigue un pre procesamiento. En éste, un tipo de ensanchamiento del histograma es realizado. Primero hay que restar el valor mínimo

de cada pixel. Después viene una corrección gamma, que es una manipulación exponencial. Debido a que es una transformación no lineal, los tonos de gris oscuros son amplificados más que los tonos claros. Para aumentar este efecto, un procesamiento de contraste sigue. El ensanchamiento del histograma es terminado con la normalización del valor máximo a 255. Estamos tratando también en usar curvelets [10], que son un desarrollo de la teoría de wavelets, pero resultó que esto no sirve en nuestro caso.

3.3 Matching

Lowe propone en su trabajo del año 1999 [2] varios métodos para hacer un matching eficiente y rápido. El problema es la gran cantidad de dimensiones del vector de las características. Un método de búsqueda en una forma inteligente todavía no es inventada. Por eso, la distancia euclidiana de todos los puntos de interés en el espacio de características es examinada. Junto con algún procesamiento para borrar equivocaciones, el número de coincidencias es calculado.

3.4 Acotamiento los puntos de interés encontrados

El programa entregaba el número de coincidencias de cada producto con la imagen obtenida desde la web-cam. Conociendo este número podemos determinar si un producto está presente o no en la imagen. Para esto se determinó un umbral, el cual si era superado por un producto, indicaba que el producto estaba presente. Después de múltiples pruebas se determinó que un umbral adecuado es de 10 coincidencias.

4 Experimentos y resultados

Los experimentos realizados buscaban determinar entre otras cosas, las condiciones de luz óptimas y la distancia a la que se debía tomar la imagen. Mucha luz era perjudicial para los resultados finales, ya que la mayoría de ellos reflectaba la luz, haciendo más difícil la detección de los puntos de interés. Como es bastante difícil estandarizar las condiciones de luz, se configuró la cámara, regulando el brillo y el contraste entre otras cosas. En la imagen 5 se muestra una imagen captada desde la web-cam con su configuración pre determinada y en la imagen 6 reconfigurada.

De ambas imágenes podemos ver que con la reconfiguración de la cámara logramos que se vean mejor los productos y sus detalles. La reconfiguración trae consigo una mejora en el desempeño del software. Al hacer pruebas bajo las mismas condiciones de luz, se encuentran muchas más coincidencias en la imagen tratada que en la original. Esta diferencia la podemos notar en las imágenes 7 y 8, donde 7 es la detección de una caja de fósforos en la imagen original y 8 en la reconfigurada.

En la imagen 7 el programa logra detectar 23 coincidencias, pero en la imagen 8 detecta 41. Además, se reducen considerablemente los falsos positivos.



Fig. 5. Imagen obtenida con la configuración predeterminada.



Fig. 6. Imagen obtenida luego de reconfigurar la cámara.

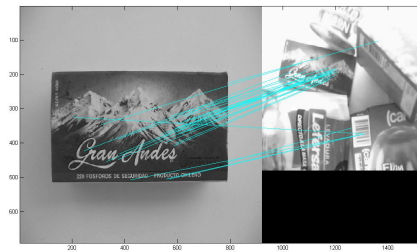


Fig. 7. Detección de una caja de fósforos utilizando una foto sin tratamiento

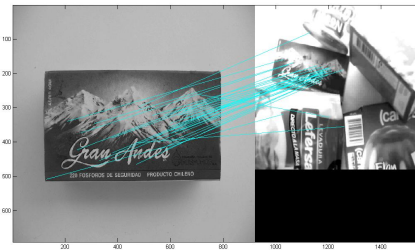


Fig. 8. Detección de una caja de fósforos utilizando una foto reconfigurada.

Para determinar la distancia óptima a la cual se debía posicionar la cámara, se hicieron múltiples pruebas. Si la cámara se encontraba muy lejos, abarcaba una gran zona, pudiendo detectar un gran número de productos, sin embargo de esta forma la detección de puntos de interés era bastante complicada, debido a la resolución de la imagen obtenida. Por esto, se buscó un punto óptimo que abarcará la mayor área posible sin una pérdida considerable de resolución.

En las imágenes 9 y 10 se muestran los resultados de buscar productos desde distintas distancias. En la imagen 9 se buscan coincidencias buscando un flan, abarcando una gran área, pero el algoritmo no logra identificar con éxito los puntos de interés, encontrando tan sólo 16 coincidencias e incluso cometiendo bastantes errores. Al tomar las imágenes de más cerca se logra mejorar el desempeño bastante, como se muestra en la imagen 10, logrando identificar 44 coincidencias y sin errores. Sin embargo el costo de este mayor rendimiento, es una disminución en el área que se abarca. Los mejores resultados se obtuvieron a una distancia aproximadamente de 50 cms. de altura, distancia utilizada en la imagen 10.



Fig. 9. Búsqueda de un flan a una distancia de 80 cms. aproximadamente.

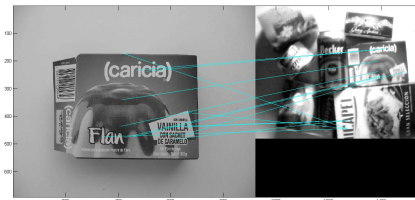


Fig. 10. Búsqueda de un flan a una distancia de 50 cms. aproximadamente.

Además como medida de desempeño se probó buscando productos que no estaban presentes en la imagen, obteniéndose los resultados mostrados en la imagen 11.



Fig. 11. Búsqueda de un producto no presente en la foto.

5 Conclusiones

El trabajo realizado tenía como objetivo detectar productos y calcular su precio final. Los resultados obtenidos son bastante buenos y eficientes, pero podrían ser aún mejores contando con una web-cam de mayor resolución a la utilizada en este proyecto (la cámara toma imágenes de un máximo de 640x480 píxeles). Aumentar la resolución de la cámara tendría como consecuencia un aumento en la efectividad del proyecto, pero traería consigo un mayor tiempo de ejecución.

El tiempo de ejecución total es de aproximadamente 30 segundos, es decir, tarda alrededor de 3 segundos en determinar la presencia o ausencia de cada producto. Considerando que sólo hay 10 productos este es un buen resultado, pero en el caso de existir miles de productos, esta solución no sería efectiva, ya que aumentaría el tiempo total.

Otra limitación de este trabajo ocurría al momento de detectar productos de la misma marca. Muchos de los puntos de interés encontrados mediante la metodología SIFT se encuentran en el logo del producto, por lo que de vez en cuando el software se confundía en estos casos.

La solución implementada detecta la presencia o ausencia de un producto, pero no se preocupa de la cantidad de veces que está presente. Este aspecto junto con disminuir el tiempo de ejecución para una gran cantidad de productos son los desafíos pendientes para trabajos futuros.

6 Bibliografía

1. Lowe, D.: Distinctive Image Features from Scale-Invariant Keypoints, Computer Science Department, University of British Columbia, Vancouver, B.C., Canada (2003)
2. Lowe, D.: Object Recognition from Local Scale-Invariant Features, Computer Science Department, University of British Columbia, Vancouver, B.C., Canada, (1999)
3. Sinha, Frahm, Pollefeys: GPU based Video Feature Tracking And Matching, Dept. of Computer Science, UNC Chapel Hill, Siemens Corporate Research
4. Frolova et. al.: Local Invariant Features, Powerpoint Presentation, http://cmp.felk.cvut.cz/cmp/courses/DZ0/2007_dzo_features.ppt
5. Patent EP0758081: Apparatus and method for spectroscopic product recognition and identification, <http://www.freepatentsonline.com/EP0758081.html>
6. Wedel, A.: Optimierung von SIFT Features zum Wiederfinden von Hausfassaden, Department of Computer Science, University of Bonn, Germany (2004) <http://www.ipb.uni-bonn.de/projects/vertiefung/SS04/vertiefer04/docs/pdf/Wedel-Optimierung.pdf>
7. Chen, J., Moon, Y.: Using SIFT Features in Palmprint Authentication, Department of Computer Science and Engineering, The Chinese University of Hong Kong, Shatin N.T. Hong Kong (2008)
8. Leo, M., D'Orazio, T., Spagnolo, P., Mazzeo, P.L., Distanti, A.: SIFT based Ball Recognition in Soccer Images, Institute of Intelligent Systems for Automation via Amendola 122/D 70126 Bari, Italy

9. Zhu, R., Yang, J., Wu, R.: Iris Recognition Based on Local Feature Point Matching, Tianjin Key Lab for Advanced Signal Processing, Civil Aviation University of China, Tianjin 300300, P.R.China (2006)
10. Candès, E., Demanet, L., Donoho, D., Ying, L.: Fast Discrete Curvelet Transforms, Applied and Computational Mathematics, Caltech, Pasadena, CA 91125, Applied and Computational Mathematics, Caltech, Pasadena, CA 91125, Department of Statistics, Stanford University, Stanford, CA 94305 (2006)
11. Moravec, H.: Rover visual obstacle avoidance, International Joint Conference in Artificial Intelligence, Vancouver, Canada (1981)
12. Harris, C., Stephens, M.: A combined corner and edge detector, Fourth Alvey Vision Conference, Manchester, UK, pp. 147-151 (1998)
13. Koenderink, J.J.: The structure of images, Biological Cybernetics, 50:363-396 (1984)
14. Lindeberg, T.: Scale-Space theory: A basic tool for analysing structures at different scales, Journal of Applied Statistics, 21(2):224-270 (1994)
15. Edelman et al.: Complex cells and object recognition. Unpublished manuscript: <http://kybele.psych.cornell.edu/~edelman/archive.html> (1997)