

Conteo de Objetos en una Imagen

Claudia Mayela Alcaraz Avendaño

Instituto Tecnológico de Celaya

claudia.alcaraz@itcelaya.edu.mx

Juana María Flores Vázquez

Instituto Tecnológico de Celaya

juana.flores@itcelaya.edu.mx

Resumen

El procesamiento de imágenes es útil en diferentes áreas, como en la medicina en el análisis de radiografías y conteo de células, en la astronomía contando los objetos (meteoritos) que puedan estar cerca del planeta y saber dimensiones, en la milicia, para detectar enemigos por imágenes vía satélite, entre otras. En este trabajo se desarrolló un sistema que permite procesar imágenes 2-D y 3-D tomadas por cámaras web, las cuales pasan por las etapas de segmentación de la imagen y la obtención de contornos y operaciones morfológicas, obteniendo de esta forma el número de objetos que se encuentran en ella.

1. Introducción

La visión es uno de los sentidos de percepción más importantes en los seres humanos aunque una incapacidad visual no impide el desarrollo de otras actividades mentales. Uno de los principales objetivos de la visión por computadora es poder diferenciar los objetos presentes en una imagen (en este caso, en una imagen digital), de tal manera, que esto logre que la identificación de los mismos sea una tarea más fácil de realizar. (Ramírez, 2004).

El procesamiento de imágenes se centra en 2 áreas principales, las cuales son:

- a) Mejora de la calidad de imágenes para la interpretación de los humanos.

- b) Procesamiento de los datos de una escena para que una computadora tenga una percepción y ejecute operaciones según sea lo extraído de la imagen.

El usar cámaras web en el conteo de objetos ayuda a poder hacer monitoreo del sistema desde cualquier otra computadora con internet, esto mediante un programa de escritorio remoto que permite operar la computadora donde se encuentra la aplicación.

El crear un sistema que analice y obtenga el número de objetos en **imágenes en 2-D**, y procese imágenes de **escenas en 3-D** es importante con el fin de determinar el número de objetos de la escena aunque se encuentren *traslapados*.

2. Metodología

El desarrollo de este sistema entra en el modelo DRA (Desarrollo Rápido de Aplicaciones), ya que es un modelo que enfatiza un ciclo de desarrollo corto. El modelo DRA utiliza una construcción basada en componentes, si se comprende bien los requisitos y se limita el ámbito del proyecto, el proceso DRA permite crear sistemas completamente funcionales en periodos de tiempo cortos (Pressman, 2002).

Cuando este modelo es usado en el desarrollo de sistemas de información, se debe enfocar en 5 etapas:

- **Modelado de gestión:** la información se modela basándose en las preguntas ¿Qué información conduce el proceso de gestión?, ¿Qué información se genera?, ¿Quién la genera?, ¿A dónde va la información? y ¿Quién la procesa?.
- **Modelado de datos:** se definen las características (atributos) de cada objeto y las relaciones entre ellos.
- **Modelado del proceso:** se crean las descripciones del proceso para añadir, modificar, suprimir o recuperar un objeto de datos.
- **Generación de aplicaciones:** el modelo DRA trabaja para volver a utilizar componentes ya existentes o crear componentes reutilizables. En todos los casos se utilizan herramientas para facilitar la construcción del software.
- **Pruebas y entrega:** Como se enfatiza en la reutilización, muchos componentes ya fueron probados, sin embargo, se deben probar todos los componentes nuevos y se deben ejercitar todas las interfaces a fondo. (Pressman, 2002).

3. Definición del sistema

El análisis de imágenes es una herramienta útil en distintos campos como la robótica, astronomía, medicina, etc.; sin embargo el procesamiento suele ser complejo ya que una imagen puede contar con varias irregularidades como tener una imagen muy oscura, una imagen con mucho ruido o que contenga objetos extraños que no deben ser considerados en el procesamiento. Para el análisis de las imágenes se debe iniciar obteniendo estas por medio de cámaras para mandarlas a procesar en MATLAB. Cuando se analizan las imágenes, estas deben pasar por distintas etapas antes de arrojar un resultado como la conversión a escala de grises y la binarización. De ahí delimitar los objetos que se encuentran dentro de las imágenes mediante la obtención de bordes y aplicar algunas operaciones morfológicas que ayuden a mejorar el procesamiento, después se etiquetarán los objetos por separado para finalmente obtener el número de objetos encontrados en dichas imágenes.

3.1. Módulo de toma de muestras

Cuando se carga el módulo de toma de muestras se abre la configuración de la webcam #1 con el objetivo de establecer los diferentes orígenes de captura ya que el sistema lleva conectado dos cámaras web.

Las cámaras muestran en tiempo real lo que se está capturando y dejan que el usuario tome las muestras en el momento que desee, después de esto se pide al usuario elegir un directorio donde se guardaran las imágenes para su procesamiento futuro y se da una breve explicación de lo que se debe hacer en este módulo.

Otra característica importante de este módulo es que permite cambiar la configuración de las cámaras web, esto es, hacer ajustes de color, contraste, cambiar la resolución de manera que se tenga la mejor calidad de imagen posible (Figura 1).

3.2. Módulo de obtención de bordes

Capturadas las imágenes a procesar, se abre el módulo para aplicar las pruebas de obtención de bordes (*inicialmente en este módulo se deben cargar las imágenes capturadas en el módulo anterior*).



Figura 1 Captura de imágenes usando 2 cámaras.

Se elige de una lista, uno de los diferentes algoritmos de obtención de bordes entre los que se encuentran (Prewitt, Roberts, Sobel, y Canny), lo que permite al usuario aplicar varios (cada que se ejecute un algoritmo diferente se muestra el resultado de la aplicación del mismo) y elegir el de su preferencia para llevarlo al módulo de conteo de objetos (se guardaran los resultados en el directorio seleccionado por el usuario en el módulo de toma de muestras) (Figura 2).

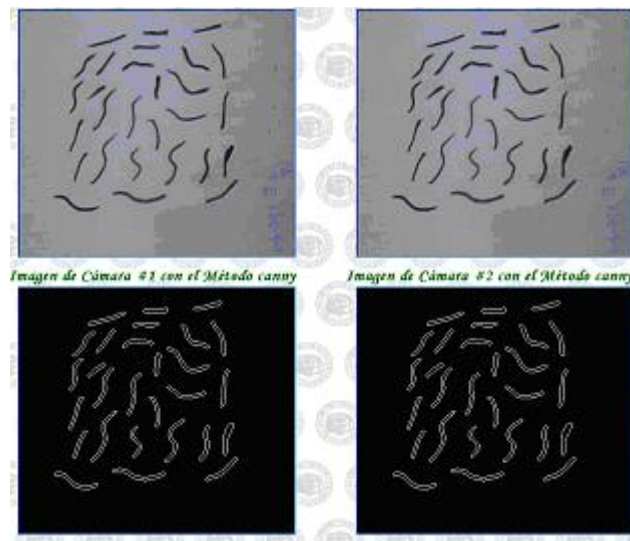


Figura 2 Resultado de aplicar el algoritmo de Canny en las imágenes.

Cabe señalar que aquí hay código relevante como la ejecución de MATLAB y el llamado de la función `imagen.m` :

```
Matlab = CreateObject("Matlab.Application")
Result = Matlab.Execute("a=imagen(" & metodo & "," & ruta & ")")
Call Matlab.Feval("imagen", 1, out, metodo, ruta)
Result = Matlab.Execute("a=imagenb(" & metodo & "," & ruta & ")")
Call Matlab.Feval("imagenb", 1, out2, metodo, ruta)
Matlab.Quit()
```

Con el código anterior se crea un nuevo objeto que es la aplicación de matlab, después se manda ejecutar la función que obtendrá los bordes y finalmente se manda a evaluar dicha función con los parámetros necesarios como son la ruta de acceso a las muestras y el método a usar, después termina cerrando la aplicación de MATLAB (Matlab Help, 2007).

3.3. Módulo de conteo de objetos

Elegido el método de obtención de bordes se procede a abrir el módulo de conteo de objetos, en el cual se hará el procesamiento adecuado que llevará a obtener el resultado deseado. En este módulo se inicia cargando las imágenes con los bordes obtenidos para procesar la imagen con algunas operaciones morfológicas que ayuden a obtener el número de objetos dentro de las imágenes a analizar.

Una característica especial de este módulo es que permite eliminar el ruido que contiene la imagen y los objetos extraños que para el usuario son considerados como basura, esto se hace estableciendo un límite en píxeles L y se eliminarán los objetos en la imagen con menos de L Píxeles ya que el usuario no los tomará en cuenta para el procesamiento (Figura 3).

3.4. Otras características

Por otra parte, aunque no es muy notable en el sistema pero no es menos importante son las funciones hechas en MATLAB, que son el principal motor en el procesamiento de las imágenes capturadas inicialmente.

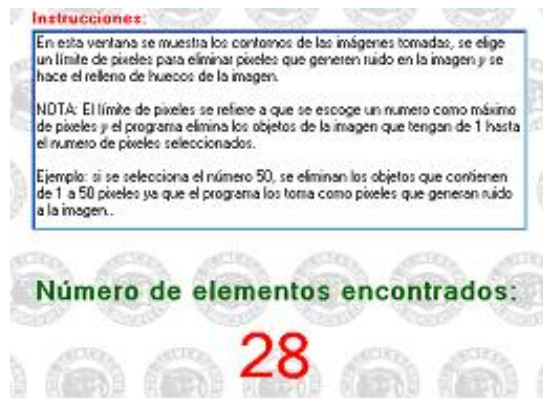


Figura 3 Resultado de procesar las imágenes con bordes.

Para poder ejecutar MATLAB desde Visual Basic .NET es necesario agregar las referencias a dicho programa. Para agregar las referencias, en visual basic ir al menu Proyecto → agregar referencia y aparecera una ventana donde se encuentran estas, ir a la pestaña de COM y ahí buscar las referencias “MATLAB Application (versión 7.0)” y la “xPC Target API Com” las cuales se abrevian como MApp Y XPCAPICOMLib respectivamente.

4. Verificación y validación del sistema

Las pruebas que se usaron abarcan desde imágenes en 2D con objetos separados y objetos juntos, hasta el análisis de imágenes en 3D con objetos separados y con objetos traslapados.

4.1. Pruebas con imágenes 2D

En estas pruebas solo se utilizó una cámara y se utilizaron frijoles sobre un fondo rojo para tener buen contraste, en la tabla 1 se presentan los resultados de dichas imágenes.

4.2. Pruebas con imágenes 3D

Para estas pruebas se hace uso de las dos cámaras, una de ellas hace la toma superior de la escena, se realiza el análisis en ambas tomas y se procede al conteo de objetos.

La tabla 2 muestra las diferentes posiciones de los objetos y el resultado del conteo de objetos con los diferentes métodos de obtención de contornos.

Tabla 1 Conteo de elementos en imágenes 2D.







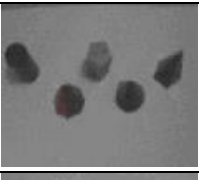




		Número de objetos				
		#Objetos	Sobel	Roberts	Prewitt	Canny
		150	150	148	150	150
		150	92	80	96	100
		29	28	27	29	28

Tabla 2 Conteo de elementos en imágenes 3D.

		Número de objetos				
		#Objetos	Sobel	Roberts	Prewitt	Canny
		5	8	5	7	5
		5	5	8	6	5
		9	9	9	9	9
		13	13	14	14	13

5. Conclusiones

Los objetivos de este trabajo se pudieron cumplir con éxito, ya que se pudo llevar a cabo el análisis de una imagen en MATLAB en cuanto a como se representa (matriz) y su conversión a escala de grises, así como la obtención de contornos y el conteo de objetos, dando como resultado un sistema capaz de contar los objetos contenidos en una imagen o una escena. Cabe mencionar que hubo ciertas limitaciones ya que se manejaron condiciones de luz y la distancia en la que se colocaron las cámaras, así como la calidad de imagen de dichas cámaras.

Un conflicto que fue de los más grandes es el almacenamiento de imágenes y resultados de matlab. Matlab cuenta con un workspace donde al ejecutar un programa se guarda información sobre las variables que se usan, pero cuando se ejecuta Matlab desde Visual Basic solo se abre una instancia de la ventana de comandos que es donde se ejecutan las funciones y no abre una ventana de workspace; debido a esto el problema se solucionó guardando en un directorio las imágenes conforme fueron procesadas y guardar en archivos el resultado del conteo de objetos.

Bibliografía

- [1] Gómez Allende Darío M. "Reconocimiento de formas y visión artificial"; Addison Wesley Iberoamericana.
- [2] MATLAB Help. Image Processing Toolbox (septiembre 2007).
- [3] Pajares, Gonzalo y de la Cruz, Jesús M. (2002). Visión por Computador 1ª Ed. Alfaomega. México.
- [4] Pressman, Roger (2002). Ingeniería del Software, un enfoque práctico. 5ª Ed. Mc Graw Hill.
- [5] Ramírez, Alonso. (2004). Reporte de Búsqueda, detección y conteo de objetos. Recuperado el 20 de Septiembre 2013 del Centro de Investigación en Matemáticas; http://www.cimat.mx/~alram/VC/ramirez_segObjDetecc.pdf
- [6] Santos, Matilde y Pajares, Gonzalo (2006). Inteligencia Artificial e Ingeniería del Conocimiento. 1ª Ed. Alfaomega. México.