CAPÍTULO 5
Aplicación del proyecto

5.1 Aplicación del sistema

Como hemos visto en Capítulo 2 existen diferentes algoritmos para el sistema de visión computacional, especialmente para enfocar el sistema y obtener las imágenes de buena calidad.

La tarea de nuestro sistema es tomar las imágenes de los tornillos de 2 mm y enfocar el sistema de manera adecuada. En la Fig. 5.1 está presentado el sistema durante toma de las imágenes.

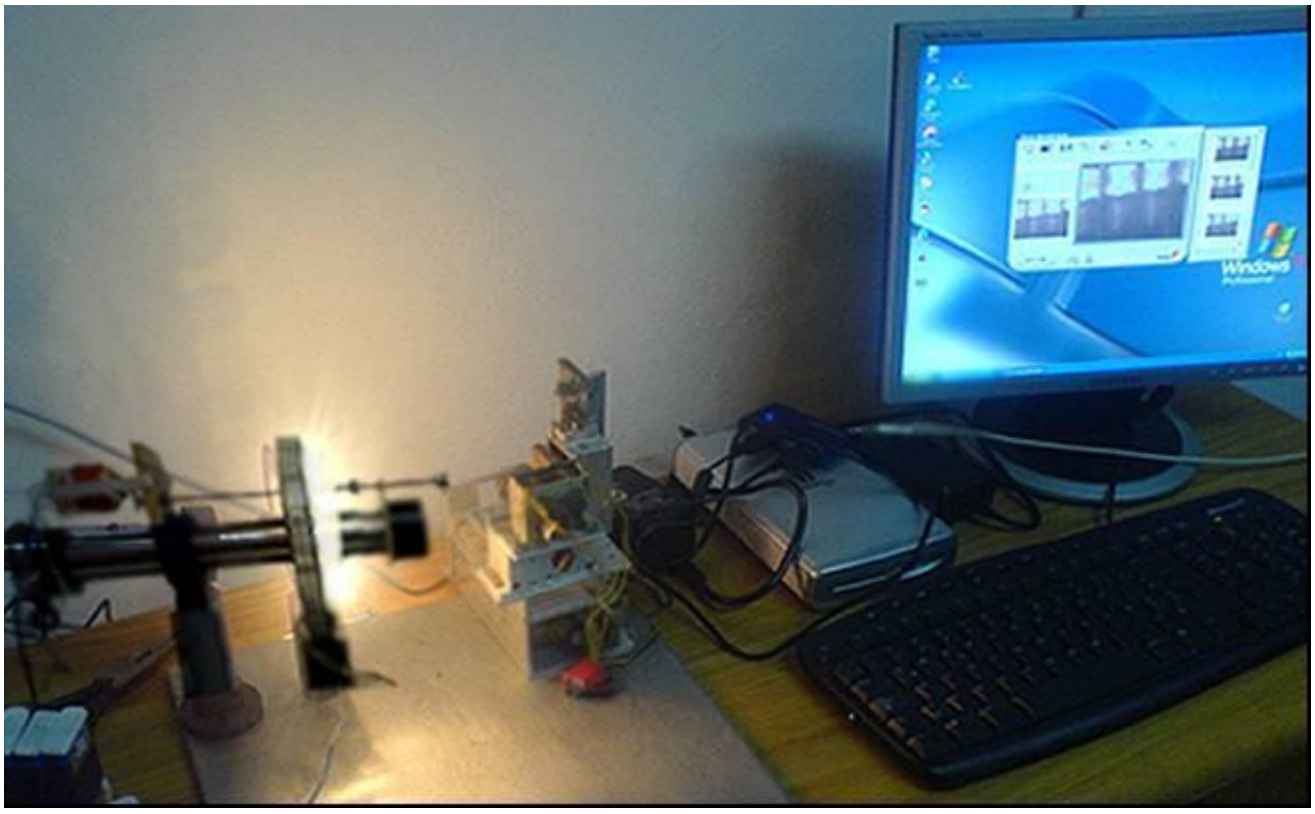


Figura 5.1 Captura de imágenes de tornillos con el sistema de visión

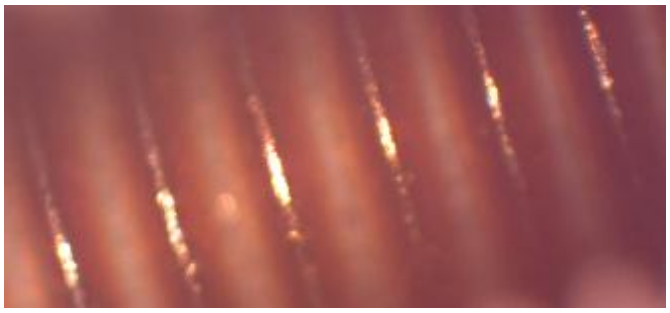
En la Fig. 5.1 se puede apreciar el sistema en su totalidad haciendo pruebas de la captura de imágenes.

5.2 Toma de imágenes con el sistema computacional

Como primer paso tomamos las imágenes con nuestro sistema con diferente distancia focal. En un inicio tenemos las imágenes muy borrosas (Fig.5.2, a), con mejor calidad (Fig. 5.2, b) y buenas imágenes con enfoque (Fig. 5.2, c). En la Fig. 5.2 presentamos los ejemplos de una cada de las tres clases de imágenes.

a

b



c

Fig.5.2 Ejemplos de tornillos con diferente distancia focal:

a - Imagen borrosa, b- imagen con mejor calidad, c – imagen enfocada

Para el sistema de visión computacional que se va a trabajar junto con el sistema desarrollado tenemos que desarrollar los algoritmos para definir un mejor enfoque.

Como primera aproximación elegimos el cálculo de los contrastes en las imágenes.

CAPÍTULO 5
Aplicación del proyecto

En Borland C++ 6 fue escrito el programa para los cálculos de los contrastes de las imágenes (Fig.5.3) (el texto de programa está presentado en Anexo 1).

En la Fig. 5.4 presentamos el ambiente del programa.

El diapasón de cambio de contraste de brillo en la imagen la dividimos por 16 intervalos. Para cada imagen calculamos el histograma de contraste. Ejemplo de trabajo del programa (imagen del tornillo con histograma de contraste) está mostrado en la Fig. 5.5. Para cada histograma calculamos suma de todos los valores de histograma. Esta suma se caracteriza en la imagen presentada al sistema.

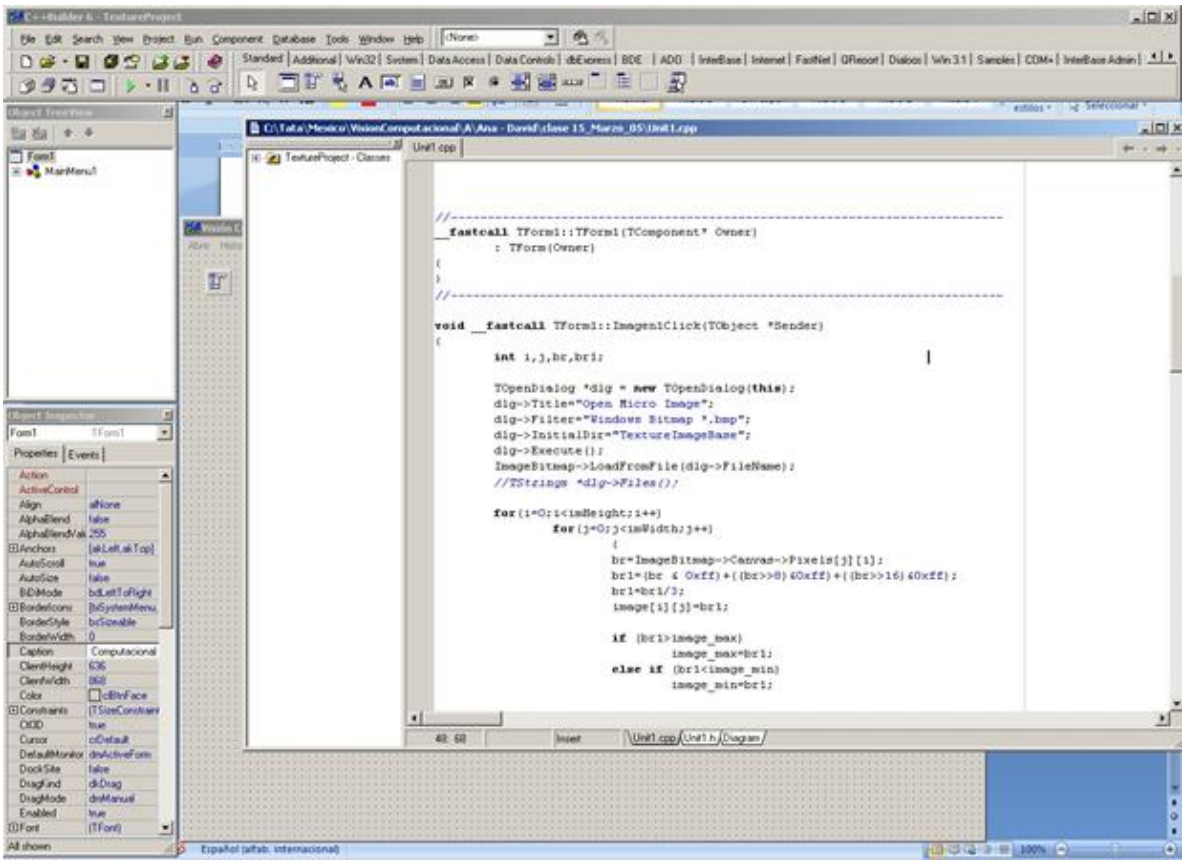


Fig.5.3 El texto del programa

En la Tabla 5.1 presentamos los resultados obtenidos para cada clase de las imágenes (los resultados son valores promedios).

Tabla 5.1.

Clase	Contraste
Borrosa	76661
Mejor calidad	87805
Enfocada	160525

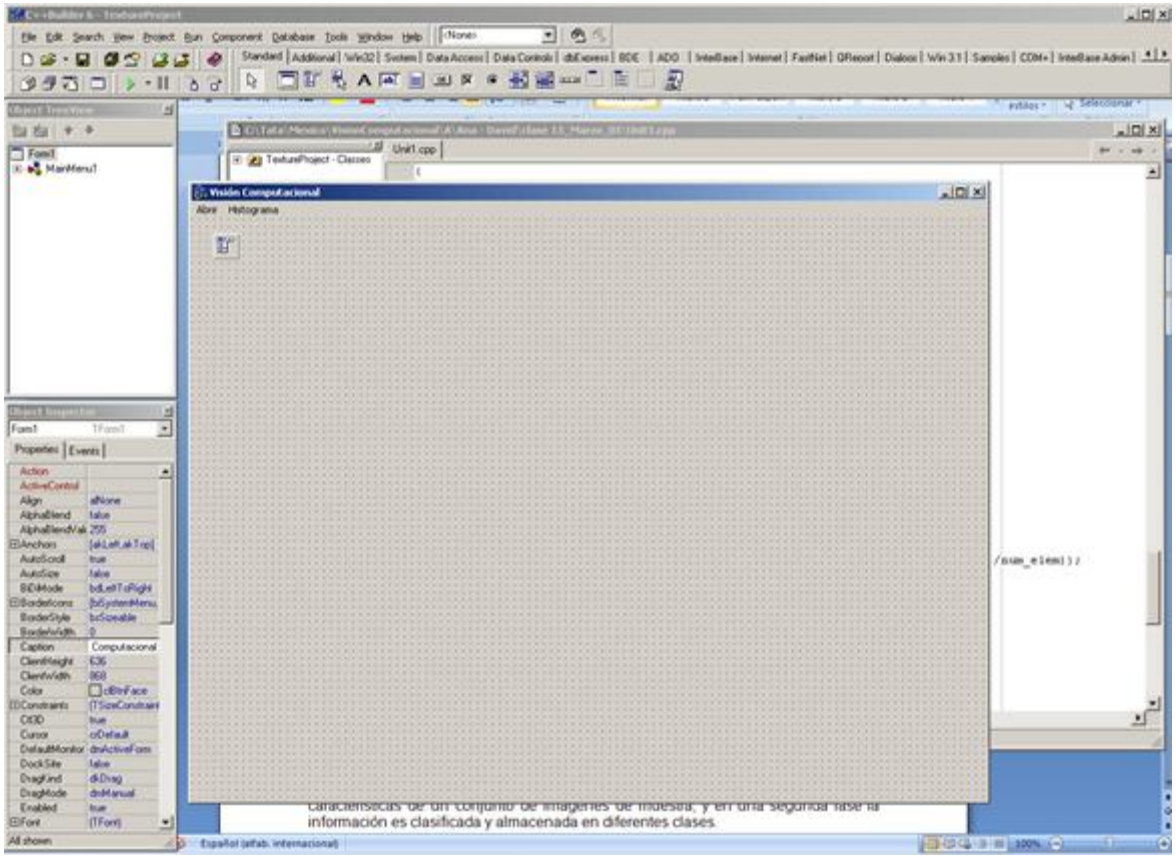


Fig.5.4 El ambiente del programa

De la Tabla 5.1 podemos ver que para la imagen enfocada tenemos un máximo de contraste. Esta característica puede ser usada en el sistema de visión

computacional para corregir la posición del lente. Entonces obtendremos una mejor calidad de la imagen cuando tengamos el máximo del contraste.

5.3 Descripción del movimiento del sistema

El control del motor de pasos puede ser realizado con siguiente algoritmo.

Paso 1. Mover el motor de pasos por X pasos.

Paso 2. Tomar la imagen 1.

Paso 3. Calcular el contraste de imagen 1 y guardar el valor del contraste.

Paso 4. Mover el motor de pasos por X pasos.

Paso 5. Tomar la imagen 2.

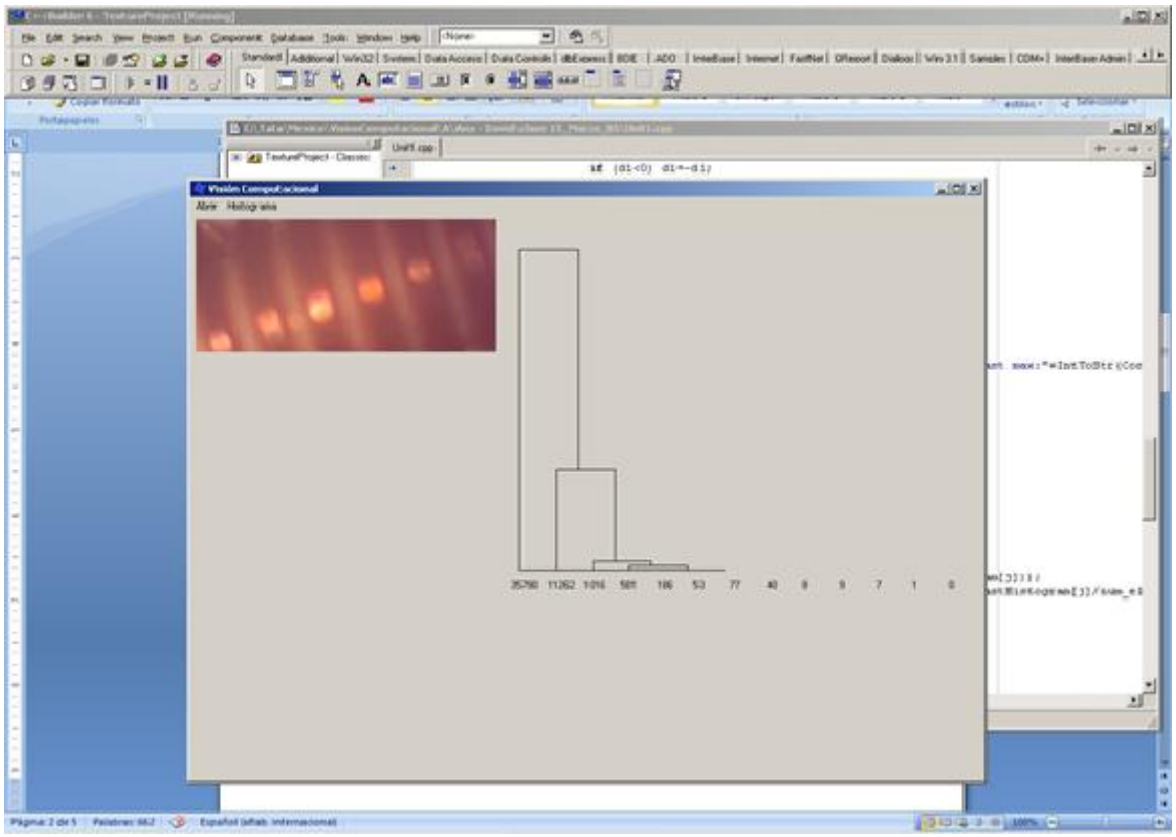


Fig.5.5 Ejemplo de trabajo del programa

Paso 6. Calcular el contraste de imagen 2 y compararlo con el valor de contraste de la imagen 1. Si el contraste es mayor que el valor previo de contraste iremos al Paso 1. Si el contraste es menor que previo se regresa el motor de pasos a la posición previa. Esta posición es la posición del sistema con una mejor calidad de la imagen.

Este algoritmo puede ser realizado en una computadora usando el lenguaje de programación C++. Este lenguaje permite usar los bloques escritos con lenguaje ensamblador que usan más frecuentemente para programar el trabajo de los dispositivos, por ejemplo, para el control de motores pasos.

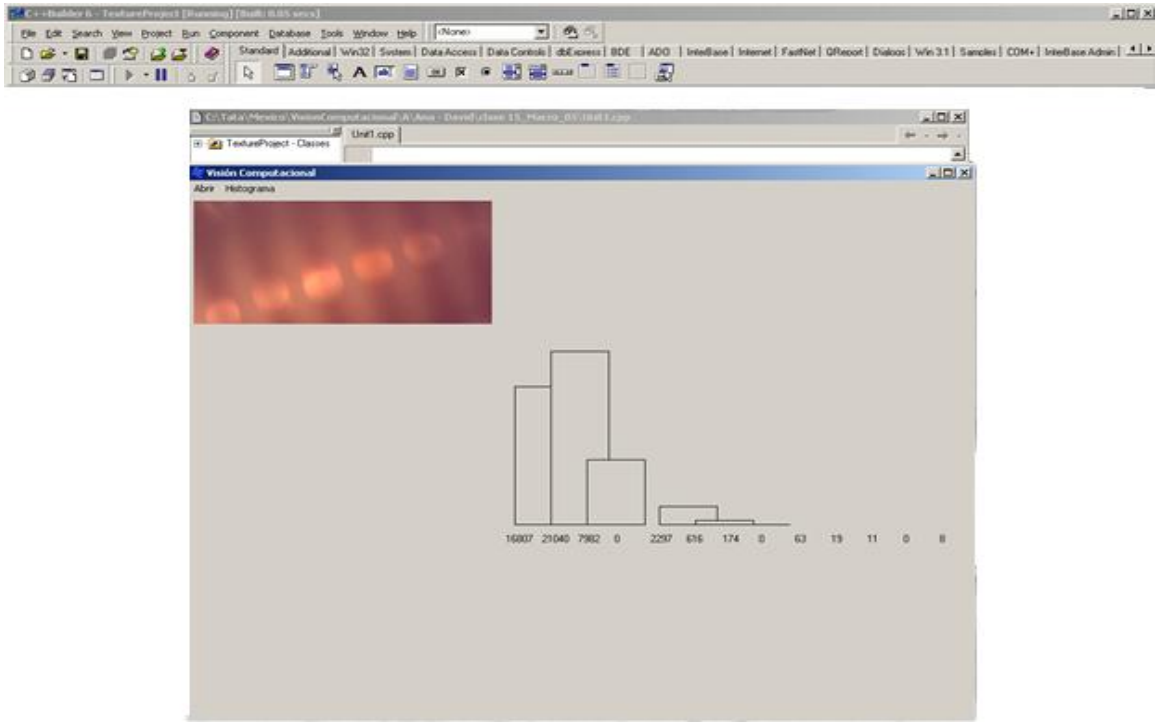


Fig.5.6. Ejemplo de trabajo del programa

Como conclusión de este capítulo podríamos decir entonces que el objetivo del programa diseñado es definir la calidad de la imagen usando el algoritmo de cálculos de histograma de contraste. Dependiendo de calidad de imagen (máximo de contraste) el sistema puede definir un movimiento del motor de pasos para cambiar la distancia focal del lente.

CAPÍTULO 5
Aplicación del proyecto

La visión computacional podría considerarse como un subcampo dentro de la inteligencia artificial y se centra básicamente en intentar expresar el proceso de visión en términos de computación. Algunas de las técnicas más utilizadas en el campo de la visión por computadora son el procesamiento de imágenes y el reconocimiento de patrones.

Un ejemplo aplicado en este proyecto es presentado analizando una serie de imágenes capturada con este dispositivo, usando unos tornillos de 2mm de diámetro.

Se desarrolló un dispositivo que se empleará para la captura de imágenes en tareas de manufactura o ensamble de microdispositivos, aplicando algoritmos de reconocimiento de imágenes.

Se obtuvieron imágenes con este sistema de visión de unos tornillos de 2mm de diámetro, obteniendo una calidad de imagen aceptable para los algoritmos de reconocimiento de imágenes.