

Reconocimiento de objetos para control de calidad mediante la descomposición en figuras geométricas

Matías Pereyra, Eduardo Destefanis

Universidad Tecnológica Nacional, Facultad Regional Córdoba

Abstract

El presente trabajo desarrolla un modelo de detección visual de formas generales en piezas industriales con el objetivo de realizar un control de calidad, a través de la detección de figuras simples y el empleo de reglas de asociación entre ellas; utilizando imágenes tomadas bajo condiciones constantes. Las reglas de asociación, la segmentación y la descripción general del objeto se realizan mediante una interfaz usuario, lo que permite establecer las características de la figura a reconocer sin la necesidad de un conocimiento especializado, y de forma intuitiva y aproximada. La detección de las características lleva asociado un valor de probabilidad, proporcionando una medida de la calidad de la pieza analizada. La asociación y el tratamiento de los resultados numéricos se basa en los conceptos y operadores de la lógica difusa. Se describen la experiencia realizada y los resultados obtenidos.

Palabras Clave

Lógica Difusa, visión computarizada, control automático, Transformada de Hough.

Introducción

En el campo “Machine Visión”, el control de calidad mediante visión artificial representa un universo creciente y en constante actualización, con innumerables ventajas en cuanto a optimización de tiempos de fabricación, facilidad de implementación y reducción de costos de producción [4]. Dentro de las aplicaciones orientadas a realizar un control sobre piezas industriales, se emplean una variedad de algoritmos que permiten analizar las imágenes para hallar ciertos elementos que brinden información sobre la eventual presencia de errores de fabricación en cada pieza.

Entre dichos algoritmos, se encuentran a menudo -debido a la geometría particular que suelen tener en común las piezas industriales-, detectores de formas geométricas sencillas tales como por

ejemplo, círculos y líneas (que pueden ser detectados a través de la transformada Hough [1]), o combinaciones y adaptaciones de esta.

Una característica de estos procedimientos, es que -en casos no triviales-, cada problema particular suele requerir: una concepción específica del algoritmo a aplicar, teniendo en cuenta la morfología específica del objeto a reconocer; los dispositivos de fijación mecánica de la pieza y de la cámara; y las condiciones de iluminación.

Por este motivo es que resulta muy poco práctico tener que desarrollar para cada nueva situación que se presente, un sistema específicamente diseñado, sensible a los símbolos, muescas, letras y/o características propias de cada objeto y al ambiente de trabajo.

Existen sistemas que abordan el problema de la generalización a través del entrenamiento con imágenes pre-clasificadas, y configurando los parámetros geométricos a analizar según los resultados deseados [8]. En [7] puede encontrarse un análisis para independizarse de los cambios de tamaño, posición, rotación e iluminación en las imágenes.

En este trabajo se propone, en cambio, emplear una serie de funciones solo para un conjunto reducido y limitado de formas básicas, para luego combinarlas entre sí mediante reglas sencillas, y analizar de esta forma figuras más complejas.

A diferencia de otros algoritmos existentes orientados a realizar esta operación en forma genérica, se pretende aquí robustecer los resultados a través de la selección de algunas características realizada por el operador del sistema [6]. Dicha selección puede realizarse mediante una interfaz

sencilla que, incorporando asociaciones lógicas probabilísticas, permita describir cada nueva situación de detección: cuantas figuras se buscarán, de que tipo, que relaciones espaciales existen entre ellas, que parámetros geométricos se deben comparar, etc. No se muestra, por razones de espacio, esta parte de la interfaz.

Usualmente en las aplicaciones industriales posibles, se persigue (luego de analizar la información) poder diferenciar aquellas imágenes que cumplan cierto criterio de calidad, de las que no lo hacen. En estos casos se definen las características esperadas para algún conjunto de elementos y se evalúa su presencia mediante un cierto umbral o parámetro establecido por el usuario.

En el presente trabajo, el proceso a nivel experimental esta planteado para una cantidad razonable de elementos básicos variados y configurables; habiéndose establecido además reglas de asociación entre las formas básicas, para definir con precisión el conjunto de elementos deseados.

El usuario deberá indicar no solo que figuras espera encontrar, sino también las zonas en la imagen que deberían contenerlas. No se considera un algoritmo de segmentación automático, sino una forma manual de realizarlo, puesto que se establece la condición a priori de que las fotos tomadas serán siempre de la misma forma y en el mismo lugar; siendo apropiado para una evaluación primaria del algoritmo propuesto y dada la expectativa de aplicarlo en procesos industriales.

Esto quiere decir que una figura (agujero, marca, muesca, letra, logotipo) debe estar donde se espera, y no es necesario rastrearla en forma global en toda la imagen, aunque si en forma local en cada región predefinida donde se espera que este presente esa primitiva.

La problemática de variaciones de orientación e iluminación, que origina la modificación de ciertos parámetros en las funciones [9] [10], queda por el momento fuera del alcance de este estudio.

Elementos del Trabajo y metodología

A diferencia de la lógica binaria, donde se establecen solo dos niveles posibles (0 o 1), la lógica difusa permite trabajar con “grados de verdad” [3], lo que es una interpretación más acertada de la realidad y como se explica mas adelante, puede asociarse adecuadamente al procesamiento de imágenes.

Una de las tantas posibilidades de este enfoque es el manejo de variables lingüísticas, utilizadas para describir ciertas situaciones (temperatura, antigüedad). Al tomar valores no numéricos, un usuario humano puede definir fácilmente el entorno, como muy frío, cálido, nuevo, viejo, etc. Se aplican luego ciertas reglas que las procesen y puedan obtener conclusiones de ellas, definiendo acciones en consecuencia.

Si bien no fue necesario implementar todas las características posibles que brinda la lógica difusa, la comprensión de sus conceptos y herramientas principales es el origen de la metodología ideada para abordar este problema.

Una red de inferencia difusa típica consiste en los procesos definidos en la figura (1).

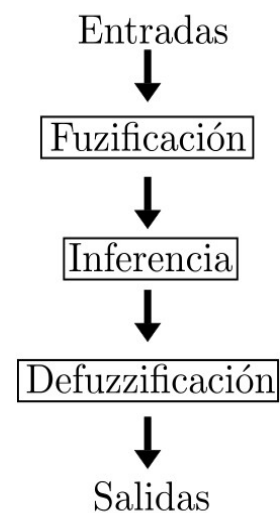


Ilustración 1: Esquema de una red de inferencia difusa

El primer paso consiste en definir numéricamente a las variables de entrada, mediante un proceso de *fuzificación*. En

este caso, se considera a cada primitiva gráfica como un conjunto difuso, y las zonas en la imagen serán las variables del universo discursivo, pudiendo pertenecer a estos conjuntos en un cierto grado.

Es necesaria entonces una forma de establecer dicha relación entre un grupo de píxeles de la imagen y una figura geométrica básica.

Se define así a una función de pertenencia μ , como aquella que mapea cada elemento de un conjunto dominio con un cierto valor continuo en el intervalo $[0;1]$, estableciendo un conjunto difuso X . En este caso, para cada región, se obtendrá un valor numérico que indique su grado de parentesco con las figuras básicas.

Esto permite definir estados intermedios entre los conceptos clásicos de formar parte o no a un conjunto, reemplazándolos por una pertenencia parcial [11].

Este concepto está directamente relacionado con uno de los objetivos principales de la investigación, el de obtener un factor de confianza en las figuras detectadas que esté directamente relacionado a su similitud con la concepción inicial de lo que se busca, imitando la percepción humana y el proceso de comparación con lo ya conocido.

Se deberá entonces proponer una forma adecuada de obtener estas funciones de pertenencia, que utilice un tiempo de cálculo razonable y que sea capaz de representar correctamente la relación entre un objeto presente en la imagen y la primitiva analizada.

Luego, el proceso de *inferencia* implica el uso de ciertas reglas de asociación para operar entre los valores de las funciones de pertenencia.

De forma similar a las operaciones entre conjuntos clásicos, se puede operar entre los conjuntos difusos. Si bien el sistema es capaz de realizar una gran cantidad de operaciones distintas, es de especial interés la intersección, por que una detección correcta completa está compuesta por una correcta detección de todas sus partes; y es una medida de la calidad del resultado la más baja calidad que lo compone. En un

conjunto difuso, se define esta operación por el menor valor de las funciones de pertenencia iniciales

$$X_a \cap X_b = \min(\mu_a, \mu_b)$$

Como cada imagen está compuesta por una sucesión de regiones, esta operación permite establecer como la calidad de la imagen completa, al menor valor que la compone.

Además, es posible generar funciones lógicas arbitrarias entre las regiones, estableciendo cuáles deben tenerse en cuenta y cuáles no, comparaciones individuales con múltiples umbrales, etc.

A continuación, la *defuzzificación* implica llevar los valores numéricos que son resultado de las operaciones anteriores, de nuevo a valores lingüísticos aplicables a la toma de decisiones.

Existirá un cierto umbral límite de calidad que deberá ser comparado con el mínimo resultado de las funciones de pertenencia para establecer si la imagen tratada cumple con las condiciones. Dicho resultado, aceptar o rechazar, consiste en la *salida* del sistema.

Uno de los principales beneficios de la lógica difusa es el manejo de los datos imprecisos, situación constante en el manejo de la visión artificial dado por el ambiente, los sistemas de captación de imágenes o la digitalización [12]. Además, el uso de reglas de inferencia difusas permite operar fácilmente entre los conjuntos para obtener la información deseada.

Para ejemplificar la aplicación y versatilidad de la idea desarrollada, se elijen dos figuras básicas combinables como conjuntos difusos: la línea y los segmentos de circunferencia (siendo los círculos un caso particular). Así, cualquier figura plana que pueda ser considerada como una combinación de líneas rectas en cualquier orientación, y arcos de circunferencia arbitrarios, se encuentra en el alcance de este sistema. Podría luego repetirse el

proceso para otras figuras, y aumentar así la capacidad de análisis.

Para realizar la tarea de detección de figuras en imágenes, es conveniente partir de las herramientas ya existentes. Se opta por una plataforma libre de visión computarizada (*OpenCV*) que posee un conjunto de funciones especializadas, además de la posibilidad de acceder al código de las mismas para modificarlas según sea conveniente.

En el marco de esta experiencia, los detectores de características gráficas de interés se basan en la Transformada de *Hough*. Configurando una serie de parámetros específicos, permiten encontrar en la imagen estas figuras y brindar los datos más relevantes de las mismas; como centro, radio, inclinación, etc. Lo que no proporcionan es un factor de calidad, es decir, con cuanta seguridad podemos afirmar que esa detección es correcta.

En correspondencia con el fundamento de la transformada, las funciones escritas en *OpenCV* analizan la imagen como una matriz de puntos [1], almacenando en un acumulador cuantos de ellos son candidatos a formar parte de la figura que se busca, y entregando la información de aquellos que superen cierto umbral. Este valor crítico (como el resto de los parámetros de las funciones) son extremadamente dependientes de cada caso en particular; tanto de las características de la imagen (iluminación, contraste, ruido), como de la figura deseada (tamaño, espesor), por lo que es necesario independizarse de estos valores.

Sin embargo, en el presente trabajo no se harán consideraciones con respecto a la imagen en sí misma o a los procesos para tratarla, solo a las figuras y su detección ¹.

¹ Uno de los pasos previos al análisis consiste en transformar la imagen en blanco y negro, en la aplicación de un filtro gaussiano para suavizar la imagen y disminuir el ruido, y la detección de los bordes a través de *Canny*, proceso dependiente de un umbral en la diferencia de intensidad de luz para hallar puntos de frontera (bordes). La efectividad de la herramienta aquí desarrollada depende en gran medida de esta detección, por lo que para cada nueva situación, se debería hallar empíricamente o desarrollar una forma automática para definirlo.

Lo que se propone para *fuzificar* las variables de entrada y obtener el valor de las funciones de pertenencia, es comparar el valor del acumulador que le corresponde a cada posible figura con la cantidad de puntos que tendría idealmente la forma buscada.

La cantidad de píxeles que forman un elemento de un cierto tamaño y grosor, son constantes, por lo que la desviación de este valor ideal es una medida de la confiabilidad de la detección. Además, pueden calcularse solo una vez para cierto rango y luego utilizarse una tabla para la comparación, resultando en una forma eficiente de obtener las funciones de pertenencia. Se utiliza la misma librería para dibujar la forma ideal y contar los píxeles que la componen (eligiéndose como valor predeterminado 1 píxel de grosor). Se utiliza *OpenCV* también para realizar el tratamiento previo en la imagen antes de la misma transformada.

Ambas funciones (la detección de círculos y la detección de líneas) se comportan de manera similar, por lo que este método es válido para las dos figuras: en el caso de los círculos, la cantidad de puntos está definida por el radio y la longitud del arco entre los ángulos inicial y final; mientras que para la línea, queda definido según el largo de la misma entre sus extremos.

En las librerías empleadas, la transformada por sí misma no devuelve el valor del acumulador en forma nativa, por lo que fue necesario modificarla, para obtenerlo y luego realizar el análisis correspondiente. No se incluye además una herramienta de detección de arcos de circunferencia.

En la figura (2) se muestran los píxeles necesarios para dibujar solo una de las primitivas mencionadas (la circunferencia), puesto que la línea recta posee una dependencia lineal entre su largo y los píxeles de su trazado, y su gráfica es trivial. La imagen solo muestra un rango limitado de píxeles para hacer énfasis en la no linealidad de la curva.

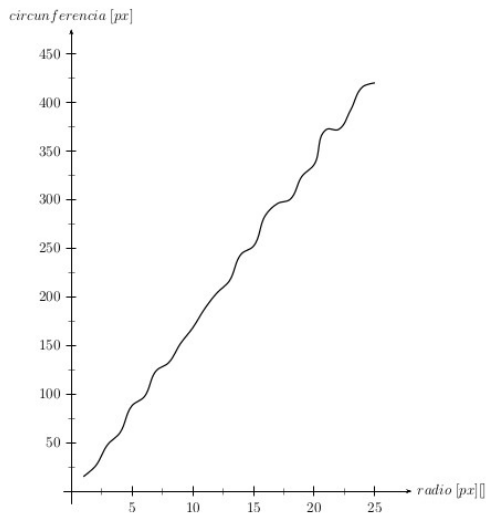


Ilustración 2: Cantidad de pixeles necesarios para dibujar una circunferencia según su radio

El proceso de combinar las figuras básicas para formar otras más complejas se implementó mediante dos tipos de reglas básicas:

- **Distancias entre puntos singulares:** Se puede asignar a un par de figuras una cantidad de puntos característicos, para analizar una distancia arbitraria entre ellos y a partir de esta, establecer otro medidor en la calidad del proceso. Una distancia nula representa puntos de unión entre las figuras, y una distancia distinta de cero, un parámetro espacial que define la distribución de los elementos.
- **Restricciones geométricas:** Se puede considerar además una comparación entre elementos propios de cada figura, por ejemplo, paralelismo entre rectas, distancia o alineación de centros de circunferencia, etc. Esto es posible utilizando los valores de retorno de las funciones, pero debe considerarse cada posible combinación de parámetros de forma particular, por lo que pierde generalidad frente al usuario.

También es posible establecer cuales regiones tienen prioridad (su confiabilidad debe superar el umbral) y cuales no (puede no superarlo), para determinar si la imagen analizada es aceptada o rechazada. Este umbral se ha considerado empíricamente para los casos de prueba, pero podría por ejemplo, definirse en función de una cierta cantidad de casos de prueba previamente clasificados para calibrar el sistema a la exigencia de calidad buscada.

Resultados

La implementación del proyecto se realizó sobre C++ con las librerías de *OpenCV* incluidas, funcionando sobre una plataforma *Linux* en una PC con 1,5GB de memoria ram y un procesador AMD Turion(tm) 64 Mobile Technology MK-36 de 800MHz. El sistema trabaja a tiempo real, es decir, luego de establecer todos los parámetros, los resultados se calculan y muestra en un promedio de 3 segundos.

A los efectos de implementar un sistema que combine los algoritmos ya citados (detectores de primitivas e inferencia difusa), se concibió la siguiente heurística sobre la cual desarrollar y operar un sistema. La misma comprende dos modos de operación, Setup y Detección:

-Modo Setup: mediante la interfaz de usuario, inicialmente se definen los siguientes parámetros

1. Cantidad de zonas a reconocer η que será equivalente a la cantidad de figuras requeridas, ya que en cada una se selecciona exclusivamente el mejor candidato y se descarta el resto. Este proceso implica la mayor relación acumulador/valor esperado.
2. El umbral mínimo de detección θ que se utilizará luego para aceptar o rechazar las imágenes. Si bien el trabajo se plantea con un umbral único, puede cargarse incluso más de uno, para realizar comparaciones individuales en cada región.

3. Regiones de interés X_k : se definen sobre la imagen base a utilizar, que servirá de referencia para las próximas detecciones. Así quedan definidas las ROI para las funciones a aplicar posteriormente (rectángulo de la figura 3). Como se considera que las fotos se toman siempre en la misma posición, las coordenadas de dichas regiones se definen una vez y se mantienen fijas.
4. Asignación de una asociación disyuntiva o conjuntiva para la primitiva a detectar en cada ROI: cada zona, se establece como de prioridad (debe superar el umbral de probabilidad difusa para que la pieza sea aceptada) o como opcional (puede no superarlo). Aquí también se puede generar una función lógica arbitraria, una comparación individual con múltiples umbrales, etc.
5. Definición del tipo de figura a encontrar y especificación de valores para parámetros requeridos para las primitivas:
 - a) Líneas: Se debe elegir un largo mínimo, para descartar resultados erróneos o no esperados, producto del ruido en la imagen. En esta experiencia se definen interactivamente.
 - b) Arcos de circunferencia: La forma de ingresar el radio mínimo esperado se realiza mediante una interfaz como en el caso anterior (figura 3). Esto es especialmente importante para la transformada de Hough de círculos, para poder diferenciar circunferencias concéntricas [2]. Además, se establecen los ángulos iniciales y finales que definen al arco (idem anterior).
6. Definición de restricciones para cada región δ : El siguiente paso,

consiste en definir las reglas de asociación para las figuras. Podemos elegir pares de puntos entre los elementos para medir una cierta distancia arbitraria, y considerar la desviación del valor esperado como otro valor de detección; o también relacionar parámetros característicos de las figuras y comparar sus distancias o similitudes, como centros de círculos, o pendientes de rectas.

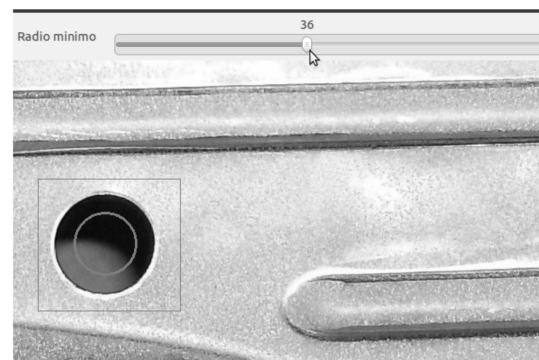


Ilustración 3: Se establecen los parámetros a través de una interfaz gráfica de usuario

-Modo Detección: Se tomaron fotos de una pieza estampada perteneciente a la carrocería de un automóvil, con una cámara de 8MP en condiciones no controladas de luz. Se utilizó una imagen inicial para cargar todos los datos, para luego analizar una serie de imágenes de la misma pieza que la primera, pero con pequeñas modificaciones y defectos para generar distintos resultados y analizar la respuesta del sistema frente a los errores en la pieza. El siguiente paso es lograr la detección según los parámetros especificados. Como ya se explicó, se modificó el código de *OpenCV* para lograr este objetivo.

- Detectar círculos completos: Era necesario obtener el valor del acumulador de la transformada de *Hough* para cada círculo detectado. Este es calculado internamente, para poder compararlo con un cierto valor mínimo (parámetro de la función al ser llamada por la

librería), pero no es retornado por la función.

- Arcos de circunferencia: Cuando la transformada considera puntos para un posible círculo, las distancias de los puntos de los bordes al centro se calculan, y se suman candidatos por cada punto que tenga un radio adecuado. Como es necesario descartar aquellos que no se encuentren entre los ángulos inicial y final, ingresados por el usuario y que definen al arco, se modificó también esta parte del algoritmo, puesto que la librería no cuenta con una función de detección ad-hoc aplicable a segmentos de circunferencia.
- Líneas: Se utilizan por separado dos funciones, la primera proporciona las rectas detectadas en notación polar y la segunda brinda los puntos iniciales y finales del segmento. Se cruzan los resultados de ambas para encontrar las líneas coincidentes y de mayor cantidad de puntos detectados. Luego, el valor del acumulador de la función (no entregado en forma nativa) se compara con el largo ideal que debería tener la recta, definida por los puntos iniciales y finales que estableció la segunda función.

El método establecido para hallar los valores de las funciones de pertenencia μ_k es a través del porcentaje del valor ideal de la figura s_k que representa el valor del acumulador obtenido h_k

$$\mu_k = 100 - |s_k - h_k| (100 / s_k)$$

Esto permite considerar tanto valores por defecto como por exceso (debido al ruido en la imagen). La función de pertenencia de las circunferencias dependen de dos variables: el radio y el acumulador de la transformada (s_k es una variable intermedia, puesto que esta definida por el radio).

En el caso de los segmentos de circunferencia y las rectas, el cálculo es análogo, considerándose como figura patrón la primitiva correspondiente para calcular s_k .

Luego, se obtienen las distancias y demás comparaciones geométricas especificadas por el usuario, según las restricciones implementadas, y se toman estos valores como si fueran otras ROI, en donde sus funciones de pertenencia están representadas por una proporción de la distancia en pixeles entre los puntos elegidos.

Se realiza la clasificación de las imágenes de prueba, definiendo cuales están aceptadas y cuales no, operándose sobre aquellas regiones establecidas como prioritarias y considerando simultáneamente las regiones de interés y las distancias como conjuntos difusos, comparando los valores de las funciones de pertenencias con el umbral θ

$$\bigcap_{k=1}^{k=\eta+\delta} X_k > \theta$$

Se muestra a continuación el resultado de aplicar el sistema a una imagen de prueba, en donde se espera detectar los errores en una pieza utilizando otra en buen estado como control.

Las figuras encontradas se muestran en color sobre la imagen utilizada, detallando además su confiabilidad individual y los datos generales de la figura geométrica, como su centro, su radio, etc.

Se definen dos regiones de circunferencia completa. Luego de realizar la segmentación manual dibujando en la pantalla las regiones de interés, se introducen interactivamente los valores mínimos aproximados de radios. De esta forma, se ahorra tiempo de cálculo al descartar aquellos círculos que no cumplen con este requerimiento.

Se detallan además los ángulos iniciales y finales que definen la circunferencia; siendo en este caso 360° completos.

En la figura 4 se muestran las imágenes para la prueba del software, en donde a) es la pieza de control y b) la pieza defectuosa.

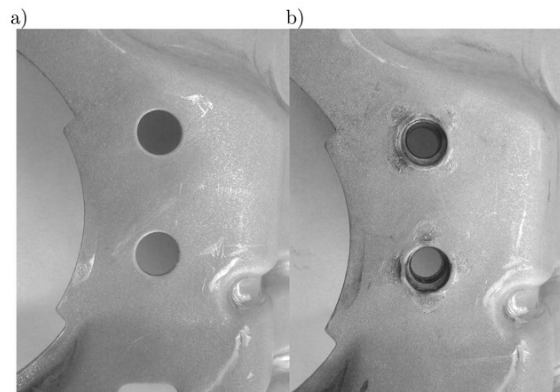


Ilustración 4: Imágenes para realizar la prueba

Para realizar el control de calidad, se establecen como parámetros: la confiabilidad individual de la figura detectada, en donde se toma como referencia al valor obtenida en la pieza sin fallas y una cierta tolerancia bajo ese número; el radio del círculo, puesto que un valor tanto por encima como por debajo significa que se encontró en esa región un círculo bien definido pero de dimensiones incorrectas, y la distancia entre los dos centros hallados, puesto que las perforaciones podrían hacerse correctamente pero en lugares erróneos.

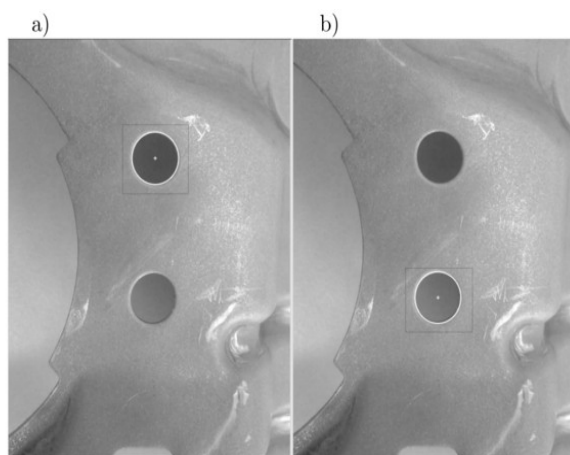


Ilustración 5: Detección en la pieza de control

Los resultados de la primera detección pueden verse en la figura 5: para a) se obtiene una circunferencia de radio 161

pixeles y confiabilidad 67,90; para b) se obtiene una circunferencia de radio 158 y confiabilidad 49,56. La distancia entre los centros es de 864,11. Los resultados de la detección en la pieza fallada se observan en la figura 6: para a) se detectó una circunferencia de radio 165 y confiabilidad 55,30; para b) se detectó una circunferencia de radio 128 y confiabilidad 66,35. La distancia entre los centros es de 863,23.

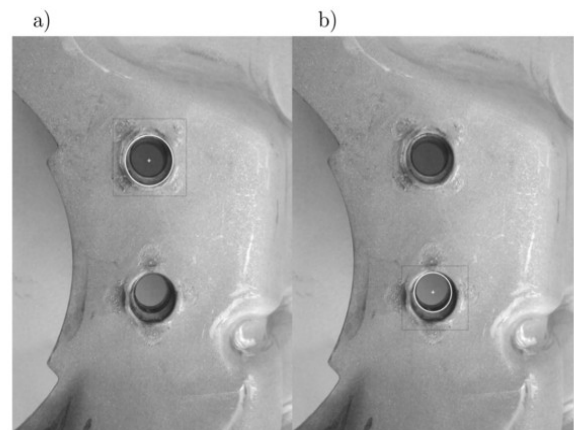


Ilustración 6: Detección en la pieza defectuosa

Discusión

Se realizó una implementación a pequeña escala de la idea propuesta. Contando únicamente con dos primitivas básicas y las herramientas de asociación de figuras, es posible descomponer una gran cantidad de figuras complejas sin la necesidad de realizar cambios en el software. El ejemplo se desarrollo sobre una pieza de auto real, y solo fue necesario el análisis de circunferencias. Pero existe la posibilidad de guardar a modo de plantillas archivos de configuraciones para construir bloques cada vez más específicos componiendo las figuras geométricas ya descriptas en el trabajo.

Si bien se utilizaron solo dos piezas diferentes, la pieza dañada posee ligeras deformaciones en los círculos así como un agregado de circunferencias concéntricas más pequeñas; detalles que deberán reflejarse en los resultados.

No se esperaba, aún para la pieza en perfecto estado, un valor cercano al %100 en la calidad de figuras obtenidas, puesto que como la referencia en el cálculo es una figura geométrica dibujada digitalmente, cualquier ruido en la imagen o deformación debido al ángulo de la foto puede generar una gran diferencia con una figura ideal y sin distorsiones. Se debe contar entonces con una serie de varias piezas de control para hallar el umbral correcto para esta pieza, puesto que las condiciones controladas en las que se opera aseguran la repetición en forma similar de las desviaciones.

El sistema fue capaz de reconocer, en la figura de arriba, el mayor círculo en ambos casos; pero la pieza deteriorada obtuvo un valor de confiabilidad menor en 12,6 puntos, por lo que es posible rechazarla eligiendo correctamente el umbral.

En la segunda región, como se elige de todas las figuras aquella con mayor nivel de confiabilidad y se descarta el resto, se encontró como mejor candidato al círculo más pequeño localizado dentro de la circunferencia original. La diferencia de radio entre este y el de la pieza de control permite agregar otro criterio para descartar la segunda pieza, aún cuando la confiabilidad individual sea alta.

El último parámetro a considerar es la distancia entre los centros de los círculos: existe una mínima diferencia entre la primera y la segunda pieza, la cual no existiría si se hubieran hallado los dos círculos que se esperaban; por lo que también es una característica que contribuye a la decisión final.

Como los resultados numéricos varían para cada nueva aplicación a utilizar, no es posible definir un umbral único para todas las situaciones. Es conveniente hallar uno distinto para cada caso, a partir de un conjunto amplio y pre-clasificado de imágenes de prueba, entrenando al sistema para que logre diferenciar correctamente todos los casos esperados. Así, se evita la difícil tarea de elegir a priori un umbral, ya

que porcentajes razonables para una aplicación podrían no serlo para otra.

Una forma de lograrlo es a través de redes neuronales supervisadas.

Conclusión

Se comprobó la viabilidad de un modelo que combine inferencia fuzzy (difusa) y detectores de primitivas con una probabilidad asociada para detectar visualmente fallas en piezas industriales.

El ejemplo desarrollado en las secciones previas es la aplicación a un caso real, en donde se pudo diferenciar evidentemente y de varias maneras las fallas en la pieza analizada.

La idea propuesta para facilitar el reconocimiento de figuras complejas a partir de la combinación de figuras simples y lógica difusa es factible: si bien no se muestra explícitamente en el presente trabajo, las pruebas realizadas por los autores permiten describir las formas de las piezas, por más complicadas que parezcan, mediante las primitivas gráficas desarrolladas y las comparaciones geométricas y espaciales que fueran necesarias.

Por lo tanto, el sistema implementado demuestra su facilidad de uso y versatilidad de aplicación, aún en presencia de imágenes obtenidas con cámaras no fijas y no profesionales, y en condiciones no controladas de luz, pero si constantes (no se preparó una iluminación artificial, pero las fotos fueron tomadas en el mismo momento del día en la misma habitación con el mismo fondo).

Las herramientas para analizar y comparar las detecciones realizadas según sus características (como la desviación de un radio o un centro de circunferencia) son tan necesarias como la misma confiabilidad de las detecciones propuesta, a la hora de reconocer errores.

La forma implementada de establecer las funciones de pertenencia es eficiente, aún cuando estas dependan de varias variables y deban ser obtenidas en tiempo real; puesto que no es a través de una función de cálculo

complejo sino utilizando una comparación en cantidad de píxeles. El sistema podría ampliarse fácilmente con un mayor número de primitivas empleando un algoritmo análogo al desarrollado, y precalcular tablas con los valores de las primitivas si fuera necesario para hacer aún más rápido el proceso; si el tiempo de procesamiento actual no satisface los requerimientos de la aplicación. Hay que notar sin embargo, que el hardware utilizado se encuentra en la actualidad ampliamente superado en la mayoría de las computadoras personales de escritorio, en unas tres o cuatro veces.

Entre las posibles actividades futuras, se considera investigar una forma de optimizar los parámetros internos de las funciones para que puedan utilizarse en imágenes en condiciones de luz y calidad totalmente variables, para que estas no deban obtenerse empíricamente cada vez en un proceso de prueba y error. El detector de bordes Canny por ejemplo, proceso vital antes de utilizar la transformada de Hough, requiere un umbral de intensidad de brillo exacto para funcionar, variando este entre una situación y otra.

Además, es posible implementar un sistema de clasificación para obtener los umbrales automáticamente a partir de los casos de prueba.

Agradecimientos

La presente investigación ha sido llevada a cabo en el marco del proyecto PID UTN 25/E139, dentro del Grupo de Inteligencia Artificial y el Centro de Investigación en Informática para Ingeniería de la UTN FRC.

Referencias

[1] Transformada de Hough.
http://en.wikipedia.org/wiki/Hough_transform

[2] G. Bradsky, A. Kaegler: Learning OpenCV Computer Vision with the OpenCV Library. O'Reilly Media, Inc. 2008.

[3] Lógica difusa.
http://en.wikipedia.org/wiki/Fuzzy_logic

[4] Machine Vision HandBook, UK Industrial Vision Association, 2007

[5] J. Tornero, L. Armesto, N. Montés, M. Mora, A. Herráez: Quality control of car-bodies based on artificial vision. European Automotive Congress 2011.

[6] Y. Motai, A. Kosaka: Concatenate Feature Extraction for Robust 3D Elliptic Object Localization. 2004 ACM Symposium on Applied Computing.

[7] D. Lowe: Object Recognition from Local Scale-Invariant Features. International Conference on Computer Vision, Corfu, Greece (September 1999), pp. 1150-1157.

[8] R. López, E. Sotter, E. Zurek: Aplicación del sistema Robot Vision PRO para operaciones automáticas de control de calidad. Ingeniería y Desarrollo, núm. 9, julio, 2001, pp. 88-97.

[9] C. Stewart: Robust parameter estimation in computer vision. 1999 Society for Industrial and Applied Mathematics, SIAM Review Vol. 41, No.3, pp. 513-537.

[10] C. Garbe, B. Ommer: Parameter Estimation in Image Processing and Computer Vision. Model Based Parameter Estimation, Contributions in Mathematical and Computational Sciences 4, 2013.

[11] R. Rojas: Neural Networks Chapter 11, Springer-Verlag, Berlin, 1996.

[12] P. Sobrevilla, E. Montseny: Fuzzy Sets in Computer Vision: an Overview. Mathware and Soft Computing 10 (2013), 71-83.

Datos de Contacto

Pereyra Matías, Eduardo Destefanis, Centro de Investigación para Ingeniería Universidad Tecnológica Nacional Facultad Regional Córdoba.
edestefanis@frc.utn.edu.ar
58010@electronica.frc.utn.edu.ar