

Visión por Computadora
Practica No.1 Parte 2
Inspección y detección de Piezas usando software de visión industrial

Entregables: Reporte en PDF de la parte 2 y 3 de esta práctica, sus códigos .vbai y algunas de las imágenes que utilizaron.

Fecha de entrega: 15 de mayo de 2021 en la plataforma TEAMS

Fecha de exposición: 17 de mayo 2021 a la hora de clase

NOMBRES DE LOS ARCHIVOS:

Reporte escrito por equipo: **Equipo#_Reporte.pdf**

Código de la práctica por equipo: **Equipo#_codigoX.vbai** (con algunas imágenes propias), sustituir la X por el número de ejercicio. Todo agregarlo en un .zip: **Equipo#_codigos.zip**

Rúbrica de evaluación en equipos (4 integrantes por equipo)

	10	9	8	7	5
Programación (40%)	Buena Ejecución, óptima codificación, Imágenes que permiten evaluar los objetivos (cada parte de la práctica). Cada integrante explica cómo funcionan sus códigos de la práctica	Código sin optimizar, Buena ejecución Imágenes que permiten evaluar 4 de los 5 códigos, (la parte 2 es obligatorio). Cada integrante explica cómo funcionan sus códigos de la práctica	Mínimos errores de programación, mínimos errores de ejecución, el escenario planteado permite evaluar con errores mínimos la parte 2. Cada integrante explica cómo funciona sus códigos de la práctica, funcionan 3 de 4 códigos	Mala programación que lo hace lento y no muy funcional. Imágenes no bien ajustadas para evaluar la parte 2 de la práctica. Cada integrante explica cómo funciona sus códigos de la práctica, funcionan 2 de 4 códigos	No hace los cálculos correctamente, mal funcionamiento de 3 o más códigos de la práctica, principalmente el de la parte 2.

Exposición (30%)	Funciona durante la entrega y cumple objetivos solicitados, Adaptación de escenario correcta, cada integrante del equipo expone un código y de forma indicada expone el objetivo 2 en equipo. Todos encienden su cámara. Cada integrante responde a las preguntas dirigidas a él. Explican y justifican el funcionamiento de su código	Funciona durante la entrega y no cumple 1 objetivo solicitado, adaptación de escenario correcto, Cada integrante del equipo expone un código y según se solicita se expone el objetivo 2 por todo el equipo. Todos encienden su cámara. Cada integrante responde a las preguntas dirigidas a él. Explican y justifican el funcionamiento de su código	Funciona durante la entrega y no cumple 2 objetivos solicitados. No hay adaptación correcta de escenario, solo 3 integrantes del equipo exponen códigos, pero si se expone el objetivo 2. Solo 3 encienden su cámara, alguien más responde a las preguntas de otro integrante. Explican y justifican el funcionamiento de su código.	Funciona durante la entrega y no cumple 3 objetivos solicitados No hay adaptación correcta de escenario para la parte 2, Solo 2 integrantes del equipo exponen códigos, pero si se expone el objetivo 2. Solo 2 encienden su cámara, alguien más responde a las preguntas de otros integrantes. No explican o no justifican el funcionamiento de su código	No Funciona durante la entrega pero cumple al menos un objetivo y los miembros del equipo explican su escena y desarrollo de la práctica
Reporte (30%)	Bien redactado, completo, y análisis funcional de la metodología de cada código, en cuanto a los ajustes que debieron hacerse para	Bien redactado, con metodología de cada código, no agrega muchos detalles de los ajustes que	Bien redactado, datos poco explicados, análisis funcional de la metodología para cada código incompleto,	Con faltas de ortografía incompleto Sin análisis del sistema, Sin diagramas de tareas o explicación a detalle en	Con faltas de ortografía, incompleto, Sin metodología, ni análisis, sin diagramas de tareas, solo MUESTRA demasiadas imágenes para

	usar nuevas imágenes. Contiene diagramas del procedimiento realizado EXPLICA y MUESTRA las imágenes SUFICIENTES para mostrar el funcionamiento de cada parte de la práctica. No agrega código de VB. Con conclusiones	debieron hacerse para usar nuevas imágenes, con diagramas de tareas alcanzadas y solo EXPLICA y MUESTRA las imágenes SUFICIENTE S para demostrar el funcionamiento de cada parte de la práctica. No agrega código de VB. Con conclusiones	en cuanto a los ajustes que debieron hacerse para usar nuevas imágenes. MUESTRA demasiadas imágenes para demostrar el funcionamiento de cada parte de la práctica y las imágenes están poco explicadas. Agrega código de VB. Con conclusiones	cuanto a los ajustes que debieron hacerse para usar nuevas imágenes. MUESTRA demasiadas imágenes para demostrar el funcionamiento de cada parte de la práctica y SIN explicarlas. Agrega código de VB. Sin conclusiones	demostrar el funcionamiento de cada parte de la práctica y poca explicación de ellas. Agrega código de VB. Sin conclusiones
--	---	---	---	---	---

COEVALUCIÓN: Equivale al factor que multiplicará la calificación final de su práctica. Esta calificación será asignada por cada integrante del equipo a cada uno de sus compañeros, deberá ser agregada al reporte individual que cada integrante entregará sobre el análisis de la óptica, de tal manera que sea anónima y solo yo tendré acceso. **Sean imparciales y realicen una evaluación profesional que refleje la aportación, motivación y desempeño de cada uno de sus compañeros de equipo, dejando a un lado los posibles problemas personales.**

Parte 1. Usando Vision Builder para inspección de piezas.

Introducción.

Uno de los más grandes problemas en la visión por computadora es la detección o la medición de patrones específicos en una pieza. Cuando las piezas a inspeccionar son pequeñas, debe considerarse una iluminación especial que garantice la mejor resolución y estrategia para búsqueda de las fallas en la pieza. De acuerdo al escenario y objetivo de la inspección, se acomoda la iluminación y posición de la pieza para realizar una metodología de programación mucho mas sencilla, garantizando los mejores resultados.

Instrucciones:

Realiza los tutoriales 2-4, 6,7 del manual de prácticas. Utiliza las imágenes compartidas en la descripción de esta práctica para que resuelvas los objetivos similares a los que se plantean en ellas. Realiza 4 códigos tal como se indica:

Código 1. Utiliza las imágenes de la carpeta **botellas** para inspeccionar que la botella tiene su tapa. Todas las imágenes deben inspeccionarse desde el mismo código. Sugerencia: revisa el tutorial de transición de estados para apoyarte en el flujo del programa si la misma metodología no te funciona para todas las imágenes.

Código 2. Utiliza las imágenes de la carpeta **pieza** para validar la presencia de los diferentes orificios de la pieza original. Toma en cuenta que la base mayor del rectángulo de la pieza mide 5.9 cm.

Código 3. Inspección de piezas: Detección de bordes y validación de piezas mediante variables y transición de estados.

Utilizando como base la práctica 7 referente a “Looping and Variables”, utilicen objetos reales a su alcance para realizar esta misma práctica del tutorial, con el objetivo de validar si la pieza contiene el número de bordes indicados por una pieza original de referencia. Para ello generen o consigan piezas que cumplan con el estándar establecido y generen o consigan una pieza que no cumpla. Se anexan unas imágenes en la carpeta **llaves**, que pueden servir de referencia, pero deben ser capturadas nuevas imágenes.

Código 4. Realizar la práctica de Segmentación color con Vision Builder para las imágenes con al menos 3 colores que enviaron en la tarea 1. Seleccionen el color de uno de los objetos para que realicen la segmentación. En la práctica de ejemplo se selecciona la manzana amarilla.

Parte 2. Detección y conteo de piezas de colores en la imagen.

Objetivo.

Programar una rutina que realice la detección de piezas, por tamaño y por colores y despliegue a la salida el tamaño, el color y un recuadro con centroide. Las piezas reales están colocadas de frente a la cámara y pasaran a través de una secuencia de diferentes imágenes. El conteo de las piezas debe ser guardado hasta el final de la ejecución de las imágenes, donde deberá desplegarse el resultado obtenido.

Procedimiento.

1. Define un número de colores (mínimo 4 distintos) y tamaños aproximados (mínimo 4 distintos), y toma distintas fotos de estas piezas en diferentes posiciones. La carpeta piezasColor contiene algunos ejemplos de lo que puedes desarrollar. El número de imágenes con fotos de las piezas es mínimo 6.
2. Establezca los umbrales necesarios para detectar los colores en la imagen.

3. Encuentren las piezas con colores similares y muestren en pantalla con diferentes mensajes o recuadros tipo *bounding box*. Además, genere variables que lleven el conteo del número de piezas encontradas con el mismo color y diámetro. El valor de la variable en el conteo puede desplegarse siempre en pantalla o solo mostrarse al final para ahorrar tiempo de procesamiento.

ACTIVIDAD FINAL. REDACTE SUS CONCLUSIONES DE LA PRÁCTICA donde analicen los resultados obtenidos, los problemas, mejoras y posibles aportaciones futuras para mejor desempeño.