
Interface para retroalimentación con sistema de visión para un proyecto dentro de fábrica virtual

Inti Arturo Contreras Vázquez, Néstor Ramírez Morales, Luis Ricardo Vidal Portilla, Gerardo Sandoval Montes

Departamento de Ingeniería Industrial y Manufactura, Instituto de Ingeniería y Tecnología de la Universidad Autónoma de Ciudad Juárez

RESUMEN

Este artículo se basa en el desarrollo de un Proyecto llamado Fábrica Virtual (FV), y que es un proyecto de colaboración entre la Escuela de Ingeniería de Milwaukee, la Universidad Anáhuac Campus Estado de México, la Universidad Autónoma de Ciudad Juárez y la empresa FESTO. En el documento se explica la forma en que los procesos de producción automatizados y sus máquinas pueden ser operados a distancia por personas con fines educativos o industriales. En concreto, se muestra la creación y desarrollo del programa para la mejora de las habilidades de los estudiantes y la aplicación de lenguajes de programación como C++ y Visual Basic 6 con el propósito de monitoreo y procesamiento de las imágenes tomadas por una cámara web montada en un brazo robótico y por medio de una red Ethernet con protocolo de comunicación TCP y la transferencia de datos con un archivo txt mediador; Esta aplicación se utilizó para un juego de ajedrez entre jugadores situados en lugares distantes.

Palabras clave: Melfa Basic IV, Fabrica Virtual, Cosimir Control, Cosimir Professional.

INTRODUCCIÓN

Este artículo está basado en un proyecto llamado Fábrica Virtual (FV), esta es una asociación la cual interactúa con las Universidades de Wisconsin (Milwaukee), la Universidad Anáhuac campus Estado de México, la compañía FESTO y la Universidad Autónoma de Ciudad Juárez (UACJ). El esquema de la fábrica virtual se muestra en la figura 1.

Esta asociación tiene como meta el seguimiento o monitoreo de procesos de producción y la creación de programas.

El objetivo es mejorar la capacidad de aprendizaje del estudiante en el aspecto

industrial, y así generar una expectativa mayor al salir al ambiente real. La FV tiene algunos proyectos para la realización de este tipo de aprendizajes como lo son:

- Juegos
- Capacitación a alumnos con demostraciones prácticas
- Fabricación de nuevas piezas por medio del ensamblado
- Simulación de procesos industriales
- Modelación de sistemas en 3D

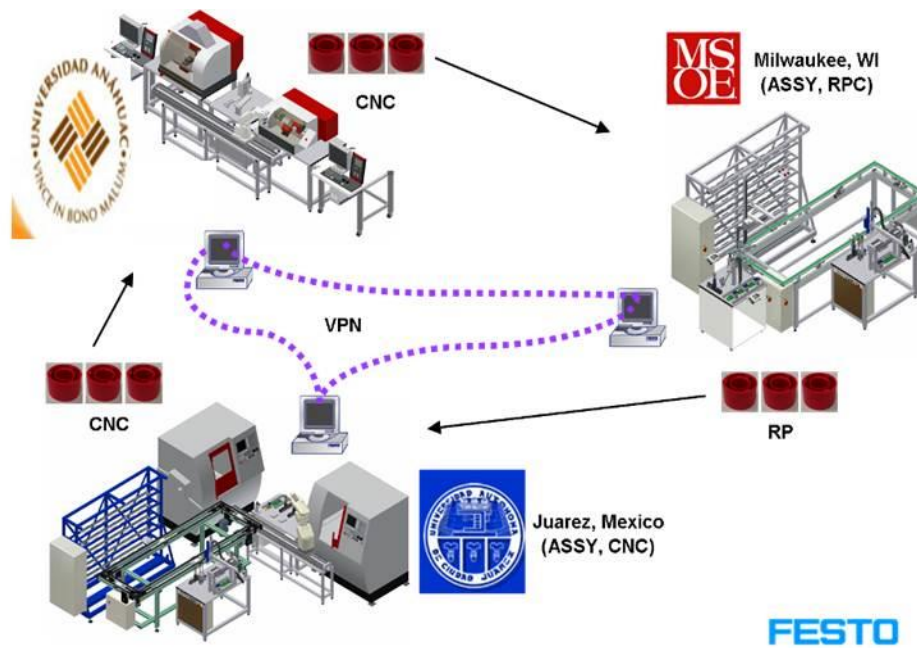


Fig. 1. Fabrica Virtual.

Dado que las tecnologías de las diferentes universidades son similares se logró desarrollar un proyecto que lleva por título “programación de un robot rv-2a para mover piezas de ajedrez con movimientos válidos de acuerdo al reglamento de ajedrez” (Mejía, 2008), el cual como su nombre lo indica logró programar el robot MITSUBISHI RV-2A para su uso en una partida de ajedrez.

El proyecto consta de un brazo robótico capaz de moverse en 6 diferentes grados de libertad mas un eje lineal con el cual se desplaza sobre toda la celda de manufactura. Su capacidad de carga es de 2kg, y puede tomar piezas de hasta 4.52 pulgadas y para ello emplea una pinza llamada “gripper”. Para este proyecto se creó un “gripper” especial para tener un mejor manejo de las piezas de ajedrez.

Para la realización de los proyectos antes mencionados la Universidad Autónoma de Ciudad Juárez cuenta con tecnología sofisticada y precisa, la cual también es capaz de guiar al estudiante en su proceso de aprendizaje ayudándolo así a la interacción con la celda de manufactura y todos sus dispositivos, la tecnología de la UACJ consta de:

- ✓ Softwares COSIMIR
 - COSIMIR® Professional
 - COSIMIR® Control
- ✓ Sistemas de producción modular (MPS) de FESTO
 - MPS Sorting (estación de clasificación)
 - MPS Processing (estación de procesamiento)

- MPS Handling (estación de manipulación)
- ✓ Sistema iCIM3000 de FESTO:
 - Modulo de visión iCIM AS/RS
 - Robot Mitsubishi Melfa RV-2AJ
 - Cámara
 - Modulo banda transportadora
 - Motores
 - Stoppers (pistones)
 - Robot Mitsubishi Melfa RV-2A
 - Fresadora CNC
 - Torno CNC
 - Estación AS-RS

Toda la celda de manufactura esta equipada con los programas antes mencionados para su eficaz interacción con cada uno de los dispositivos de la celda.

METODOLOGÍA

En este apartado se describirán las metodologías utilizadas en el diseño, manufactura y ensamblaje de la estructura que sostendrá la cámara a utilizar en este proyecto, así como la modelación de el robot Mitsubishi dentro del software COSIMIR® Professional para evitar algún tipo de colisión. Además de la descripción del algoritmo comparativo en lenguaje C#, para luego ser acoplado al programa trabajado en Visual Basic y así generar la retroalimentación.

Método

En la figura 2 se desglosa por pasos la metodología que se implementó para la realización de este proyecto se pretende iniciar con un análisis para el diseño de la estructura, después continuar con el trabajo

de fabricación de las piezas para luego ser ensambladas. El ensamble se dividió en dos partes, una es el *perfil-plataforma* (estructura) y la otra la es la *cámara-plataforma*.

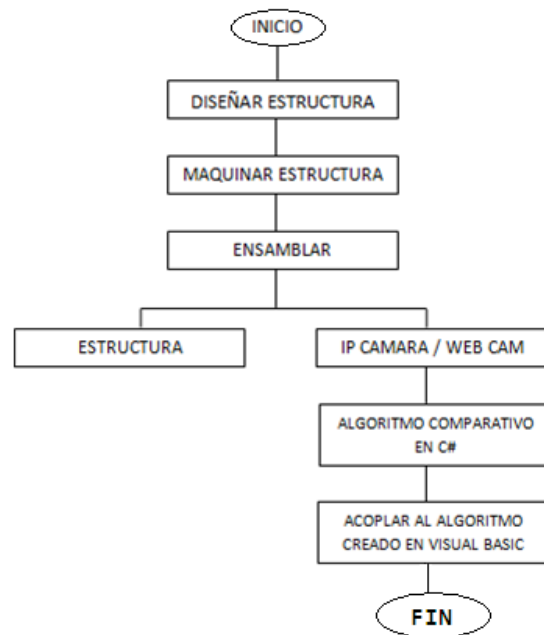


Fig. 2. Método de desarrollo

En el siguiente paso es donde se encuentra toda la información acerca de los algoritmos de comparación, algoritmo para manipular el video y los algoritmos de acoplamiento a Visual Basic.

En las siguientes secciones se analizara a fondo el diseño, el maquinado y el ensamblado descrito previamente.

Diseño de la estructura

Como se mencionó anteriormente se necesita construir una estructura de algún material resistente y fuerte, ya que esta estructura se busca que sea ajustable.

Esta característica es el principal argumento que se tomo en cuenta para el diseño de la estructura, por que al tratarse de un sistema de visión requeríamos de la mejor perspectiva para así eliminar agentes externos a la imagen.

Se realizó un análisis que nos llevo a realizar diseños usando un software de diseño para el ajuste de las dimensiones dentro de la celda de trabajo. Desde la altura del perfil, hasta la distancia desde la base del perfil hasta el eje lineal del robot manipulador.

Se podría realizar una modelación dentro del software COSIMIR® Profesional para evitar algún tipo de colisión de los movimientos del robot con la estructura de la cámara.

Maquinado de la estructura

Para el maquinado de estas piezas se hizo uso de la maquinaria que esta dentro del laboratorio de manufactura dentro de la institución, y se hicieron uso de broca, machuelo y sierra eléctrica.

Se decidió por aluminio extruido como el material principal ya que tiene unos especies de canales que puede ser perfectamente utilizados como los “rieles de ajuste” para que la altura donde se ubicara la cámara sea ajustable. Fue cortado a la medida requerida por medio de la sierra eléctrica y se removió todas las rebabas fundidas quedando listo para su ensamble con la mesa de trabajo.

Ensamblaje de la estructura

Para el ensamble de la estructura se dividió en tres partes, la sujeción del perfil con la plataforma, el acople del perfil con la

mesa de trabajo y para finalizar el enlace de la cámara con la plataforma. A continuación se detalla la manera en que se ensamblaran las partes de este proyecto.

Para sujetar la plataforma al perfil se perforo el acrílico utilizando una broca y una velocidad considerada para no quebrar el material, se utilizo un machuelo para hacer el roscado en el aluminio y poder sujetar la plataforma de acrílico con un tornillo, teniendo la estructura completa se buscara la mejor opción para la localización de la cámara.

Para la sujeción del perfil con la mesa de trabajo se utilizara un par de “eles” que sujetaran provisionalmente al poste contra la mesa por medio de tornillería estandarizada, ya que no puede ser una estructura permanente por tratarse de un espacio educativo.

Para la colocación de la cámara se utilizo acrílico, se corto un pedazo considerable para utilizarlo como plataforma ubicando la cámara por encima de ella y utilizando un tornillo para la sujeción.

Así como la cámara, el poste (aluminio extruido) y la plataforma (acrílico) serán ajustables, es decir, que se podrá modificar su altura, su longitud, el zoom y la angularidad en la imagen de la cámara.

Herramientas

En esta etapa del documento se describe las herramientas de programación utilizadas en el presente proyecto, es decir, software utilizados, algoritmos creados y código fuente desarrollado.

Algoritmo creado en C# utilizando Visual Studio

Después de haber analizado las opciones de software y lenguajes de programación, se llegó a la conclusión de que se trabajaría dentro de un ambiente de programación llamado Visual Studio (VS) y después se analizó su acoplamiento al programa AJEDREZ creado en Visual Basic (VB).

Este software facilita el trabajo de manipulación de la cámara web por medio del lenguaje de programación C# creando una interfaz amigable para el usuario. Así como la facilidad para organizar cada uno de los elementos del proyecto.

Se dividió en tres partes, una es el código de visualización de la cámara y la segunda la del algoritmo de comparación de imágenes.

Se inició el trabajo construyendo el código que fuera capaz de manipular la cámara dentro de la interfaz para después iniciar con la captura de las imágenes. Así como un menú desplegable con todas las cámaras que estén conectadas a la computadora.

La segunda etapa fue la de crear un algoritmo que fuera capaz de analizar pixel por pixel de cada una de las imágenes, obteniendo los valores RGB de cada imagen y analizándolos en un rango de color preestablecido.

La tercera parte del proyecto es la de juntar ambos algoritmos en una sola interfaz y creando un ambiente amigable de uso para el usuario.

Acoplamiento en Visual Basic

En la presente sección se describe la idea principal para el uso de la aplicación creada en C# y su acoplamiento en el programa de Visual Basic ya existente, utilizando un archivo .txt como un mediador entre ambos programas para su interacción.

Es decir, VB al correr su programa por medio del botón de “iniciar conexión” envía la señal hacia el archivo de texto leyendo la primer línea contenida en el, la cual será recibida por Visual Studio y será la señal que necesite el programa para correr el analizador de imágenes (algoritmo comparador) y tomar la primer imagen.

VS regresará la señal hacia el archivo de texto saltando a otra línea, y enviando hacia visual Basic la acción tomada.

VB remitirá la segunda señal por medio del botón que envía la coordenada a donde se dirigirá el robot por la pieza, y la señal hacia el programa analizador será detectada de la misma manera anterior pasando por líneas escritas dentro del .txt.

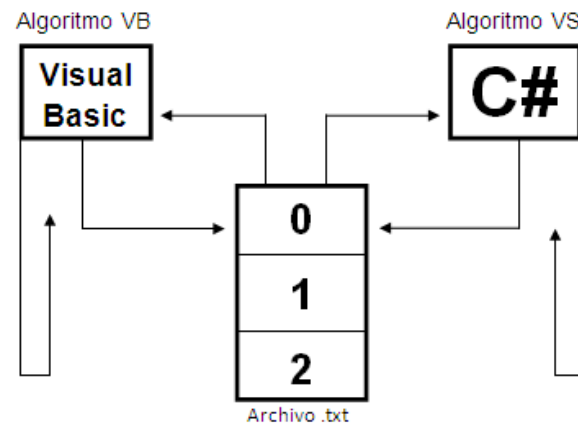


Fig. 3. Esquema de acoplamiento.

RESULTADOS

A continuación se muestran los resultados obtenidos para este proyecto. Se

explican las modelaciones, las estrategias de maquinado y ensamble. Así como los resultados obtenidos en las imágenes comparadas del algoritmo creado.

Resultados del diseño de la estructura

Como se mencionó anteriormente se requirió del diseño de una estructura metálica capaz de soportar una cámara, se inicio con el desarrollo de una modelación en un software para poder solucionar el primer problema, la ajustabilidad.

Se realizó un diseño que llevo a crear planos usando *Solid Works* para el ajuste de las dimensiones dentro de la celda de trabajo. Se diseño el perfil de aluminio, la plataforma de acrílico y la “ele” metálica para la sujeción.

En las figuras 4-7 se muestran los dibujos de las modelaciones:

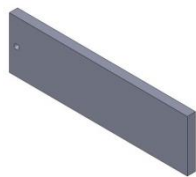


Fig. 4. Plataforma.

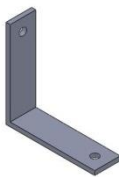


Fig. 5. “Ele”.

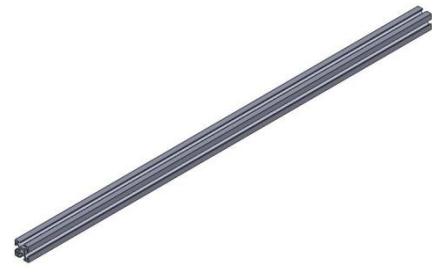


Fig. 6. Perfil (poste).



Fig. 7. Ensamble total

Resultados del maquinado de la estructura

Para la fabricación de las piezas que se diseñaron anteriormente, se dispuso del laboratorio de manufactura de la institución. Se utilizaron 3 brocas, una para la perforación tipo “through all” en la plataforma para unirlo con el perfil y otra del mismo tipo para la unión con la cámara. La otra broca se utilizo en el perfil de aluminio perforándolo una pulgada hacia adentro en la superficie, así como la utilización del machuelo para poder sujetar la plataforma. Se utilizo pintura tipo spray para dar un mejor acabado a la plataforma y al perfil. En la fabricación de la “ele” solo se requirió la pintura negra para pintarlo ya que este ya existía con los parámetros que se requerían.

En las figuras 8-10 Aquí mostramos las imágenes del material fabricado:



Fig. 8. Plataforma.



Fig. 9. "Ele".



Fig. 10. Perfil soporte.

Resultados del ensamble de la estructura

Aquí se muestra el resultado de las anteriores secciones, es decir el ensamble total de los 3 elementos maquinados y diseñados con anterioridad. Todo el trabajo realizado esta descrito en su totalidad, por lo tanto a continuación se mostrara la imagen de la estructura en su posición dentro de la celda de manufactura

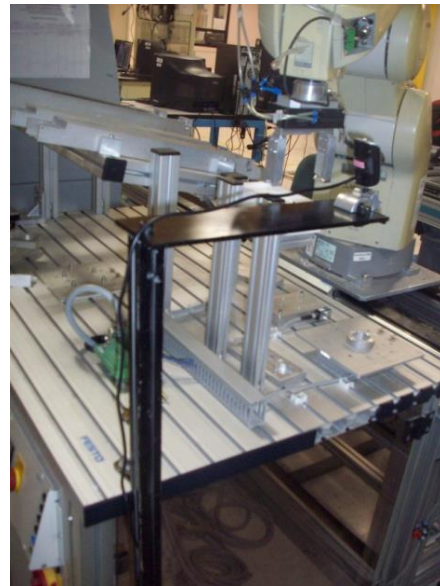


Fig. 11. Ensamble total.

Resultados del algoritmo en C#

En este punto se describe la aplicación grafica creada, así como su algoritmo para comparar las imágenes tomadas por la cámara.

Esta aplicación esta dividida en cuatro partes principales, la primera de ella es la pestaña de configuraciones, donde se creo un menú desplegable en el cual al abrirse la ventana muestra las cámaras que estén conectadas, siendo la que esta predeterminada en la computadora como la principal y la que se usará en la toma de las imágenes, la segunda parte de este proyecto

es la zona de acciones donde se puede observar un pequeño cuadro donde se carga el video que siendo mostrado por la cámara, en la tercer parte se encuentra la zona de las imágenes, es una parte donde se encuentran los frames que serán analizados para su evaluación, la cuarta y ultima parte es la zona donde se encuentra el status del juego es decir si se encuentra moviendo (el robot) o si ya regreso a su posición inicial. A continuación en la figura 12 se puede observar a detalle.

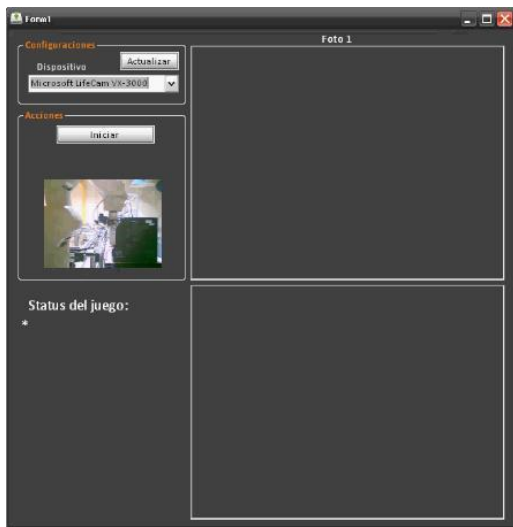


Fig. 12. Interfaz grafica del algoritmo de comparación.

Para inicializar con la aplicación se debe dar clic en el botón de Iniciar Conexión de la aplicación con el tablero de ajedrez simulado, esto abrirá en una ventana nueva la interfaz grafica con el algoritmo de comparaciones. Al abrirse esta ventana, la cámara que haya sido detectada como predeterminada será la que se cargue en el recuadro de acciones, ya que tardara unos dos segundos en abrir todo el lente de la cámara es por eso que se decidió que la primer imagen se capturara justo cuatro

segundos después de que se iniciara con la conexión al robot, para permitir a la cámara tomar la imagen inicial justo como tomara la segunda, esta imagen se quedara guardada durante todo el juego siendo la imagen referente para comparar. En la Figura 13 se observa la imagen y el video.

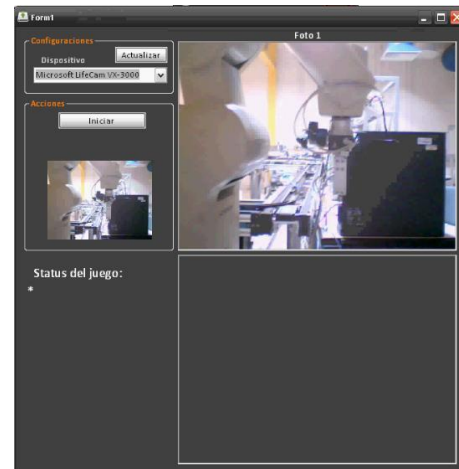


Fig. 13. Imagen uno tomada.

La segunda imagen se mostrara 4 segundos después de pulsar el botón Enviar, botón con el cual se envía la coordenada a donde se moverá el robot. Este tiempo fue decidido mediante el tiempo de respuesta del robot para “ir-venir” en un movimiento normal de pieza, el tiempo que tarda es de 6.67 segundos y para un movimiento en cual se come una pieza tarda alrededor de 14.87 segundos. Para este movimiento se presenta una variante ya que al enviar la coordenada el robot toma la pieza que se va a comer y la lleva la posición 65 para luego regresar a home, en esto tarda 7.10 segundos y dura en esta posición hasta el segundo 9.05, es decir, casi 2 segundos mas antes de salir a mover la pieza faltante, tardando así un total de 14.87 segundos. Por lo tanto el problema que se presento fue que la segunda imagen no debía tomarse durante los casi dos

segundos que el robot esta en home (entre 7.10 y 9.05), tomando en cuenta que al tomarse ambas imágenes el algoritmo de comparación tarda 2 segundos aproximadamente en deliberar si existe compatibilidad de patrones o no. A continuación se mostrara una tabla de las comparaciones con sus tiempos, así como un diagrama de flujo que se puede observar en la Figura 14.

Ejemplo:

| Imagen | Segundos al que toma la imagen | Compatibilidad de patrones |
|------------|--------------------------------|---------------------------------|
| Imagen 2.1 | 4 segundos | No existe compatibilidad +2 seg |
| Imagen 2.2 | 10 segundos | No existe compatibilidad +2 seg |
| Imagen 2.3 | 16 segundos | Existe compatibilidad |

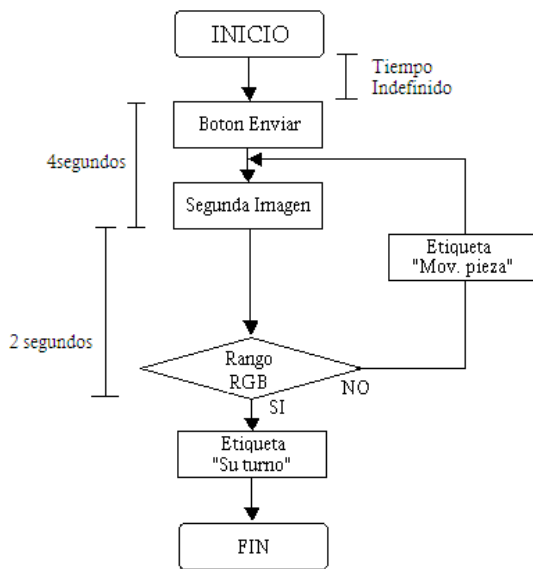


Fig. 14. Diagrama de flujo de tiempos de la segunda imagen.

Al tener ambas imágenes capturadas, como ya se menciono anteriormente el algoritmo tarda aprox. 2 segundos en exponer sus resultados, esto se debe a que todos los pixeles de ambas imágenes son

analizados, tomando en cuenta que cada pixel tiene tres valores que son rojo, verde y azul (RGB) y que van desde 0 a 255.

Al haber analizado cada imagen se obtiene el valor de los dos rojos y se compara con un rango de -80 a +80, si la diferencia de ambos valores no entra en ese rango de color nos envía un error que se va guardando en un contador y al momento de desbordar la cantidad de 2000 mil errores se envía la señal en forma de texto con la leyenda “Moviendo pieza” la cual seguirá mostrándose hasta que los errores sean menores a 2000 cambiando así la etiqueta a “Su turno”. Como se observa de color verde en la Figura 15.

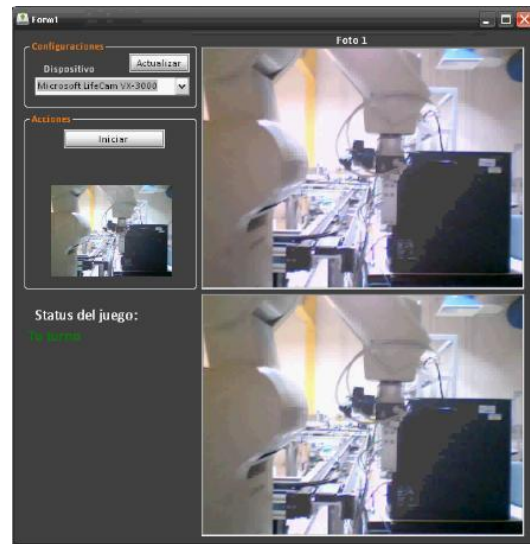


Fig. 15. Algoritmo en uso

Resultados del algoritmo en C# acoplado a VB

En la figura 16 se muestra el diagrama de flujo, donde se describe ampliamente el método de acoplamiento y sus resultados, desde los tiempos de reacción hasta el seguimiento de cada aplicación sobre el archivo de texto. A

continuación se muestra la imagen y su descripción teórica.

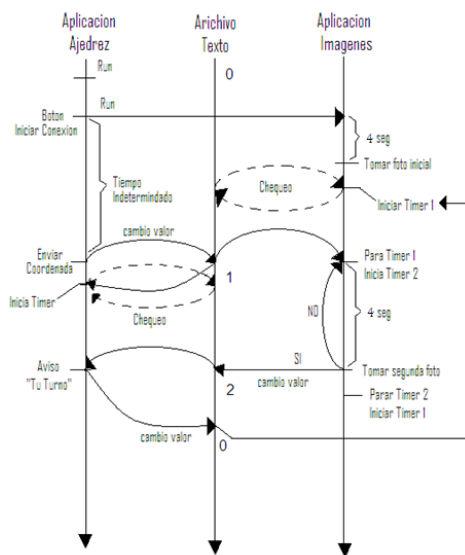


Fig. 16. Esquema de acoplamiento final a Visual Basic.

Para iniciar el proyecto se corre la aplicación con la interfaz de ajedrez, para correr la aplicación de la comparación de las imágenes se debe dar en el Botón Iniciar Conexión, situado en la parte superior del tablero de ajedrez representado con el número 1 en la figura 17. Seguido se debe de dar click en el botón comandos de inicio para verificar la conexión con el robot y sus valores de regreso. Al estar ya conectado se corre el programa comparativo, es decir, se muestra un preview del video en vivo. Dos segundos después se toma la primera imagen, la que será del robot posicionado en home.

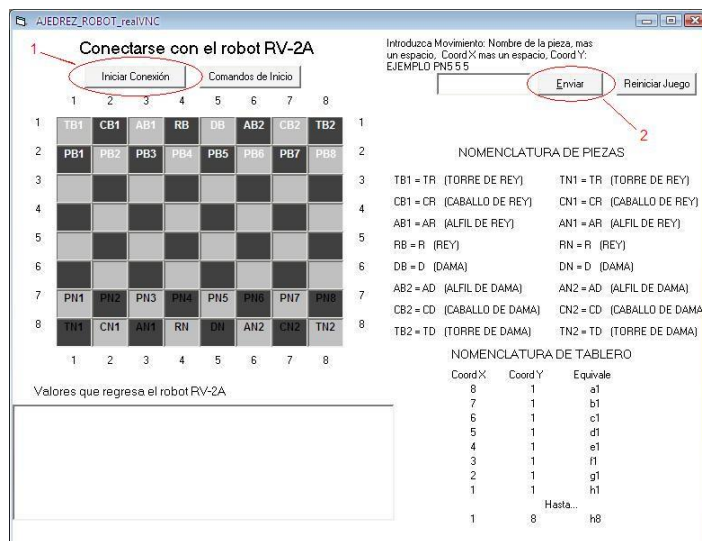


Fig. 17. Acoplamiento Gráfico.

Para continuar, se debe iniciar el juego y para ello se da una coordenada valida de juego de ajedrez dentro del cuadro de texto (ejemplo. PN5 5 5) y se da un click en el botón de enviar (desde la primer

imagen hasta la coordenada está determinado como un tiempo indefinido).

El archivo de texto se inicializa en 0, y al momento de enviar la coordenada valida, el programa lo reescribe y coloca un 1; como

ambos programas están realizando un chequeo simultaneo, el programa en C# lee el cambio de valor y realiza la acción, la cual es la toma de la imagen cada 4 segundos hasta encontrar una paridad casi total de los colores de los pixeles, al hacerlo

correctamente el archivo de texto vuelve a escribirse y cambia su valor a un 2, de lo contrario seguirá tomando frames cada 4 segundos hasta hacerlo encontrar la igualdad esperada.

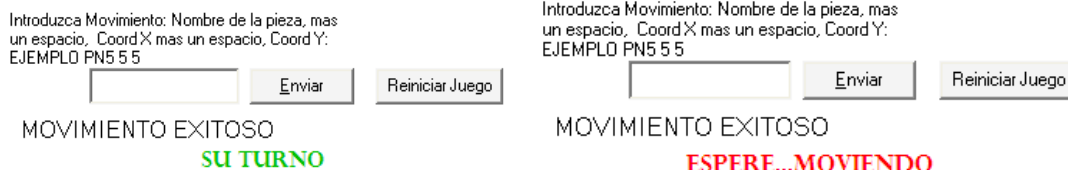


Fig. 18. Etiquetas de valores en forma de texto.

En la figura 18 se presentan las etiquetas que se muestran en la interfaz de ajedrez en forma de texto y como se puede observar mientras se siga tomando la segunda foto, en la interfaz del ajedrez aparece la etiqueta mostrando “Espere...Moviendo” sin poder realizar algún movimiento hasta no detectar al robot en home. Al cambiar el valor del archivo de texto a 2 el programa Ajedrez realiza su movimiento, el cual es cambiar el texto de la etiqueta a “Su turno”, con esto se puede volver a mover una pieza así como el valor del archivo vuelve a cambiar ahora a 0 iniciando el timer que corre el chequeo que se realiza continuamente.

El objetivo de este proyecto fue integrar un sistema de visión a una celda de manufactura, para uso exclusivo del brazo robótico en una partida de ajedrez.

Este proyecto se cumplió al 100% gracias a un algoritmo creado en C# y acoplado a Visual Basic, el algoritmo comparo las imágenes que se guardaban satisfactoriamente para luego retroalimentar sus resultados a manera de texto en la interfaz grafica del ajedrez.

CONCLUSIONES Y TRABAJOS A FUTURO

En esta última etapa del artículo se describen los objetivos o metas que se lograron realizar y tambien las recomendaciones que se observan en este proyecto, es decir, los objetivos que no se alcanzaron a cubrir, o bien aquellas metas que se pueden mejorar o reemplazar.

Al igual, para contribuir con el algoritmo de comparación se creo una estructura capaz de ser manipulada en su toda extensión ya sea su eje “X” o su eje “Y”, es decir, se construyó ajustable a diferentes tipos de cámaras (ya que cámara tiene propiedades de imagen distintas).

Existen un buen número de trabajos a futuro en este proyecto, a manera de recomendación se puede iniciar con la parte de manufactura la cual no se entro en mucho detalle ya que se centro en el trabajo de programación y la estructura no tiene el suficiente trabajo.

Se podrían revisar los tiempos de los tiempos en la aplicación en C#, para hacerlo mas eficiente y preciso, para eso se tendría que hacer unas modificaciones en el código de Visual Basic (Ajedrez) asi como en C#.

Se podría añadir líneas de programación para la posibilidad de antes de iniciar el juego se elijan si se quiere jugar con piezas blancas o con piezas negras y así se puede controlar al jugador, permitiendo solo hacer un movimiento por turno.

REFERENCIAS

Mejía, G. (2008). Programación de robot rv-2a para mover piezas de ajedrez con movimientos

válidos de acuerdo al reglamento de ajedrez. UACJ: Tesis, 2008

Ollero Baturone, A. (2008). ROBOTICA: manipuladores y robots móviles. Barcelona, Alfaomega.

Borger, A. (s.f.). Morfología del robot. En: http://www.industria.uda.cl/Academicos/AlexanderBorger/Docts%20Docencia/Seminario%20de%20Aut/rabajos/2004/Rob%C3%B3tica/seminario%202004%20robotica/Seminario_Robotica/Documentos/MORFOLOG%C3%8DA%20DEL%20ROBOT.htm

Koningshartman. (s.f.). Short introduction into 3D Simulation and Offline Programming of robot-based workcells with COSIMIR®. <http://www.koningshartman.com/NR/rdonlyres/DD76487B-95F3-4F78-9458-68986040EE4D/0/COSIMIRGettingStartedE.pdf>