**INFORMES DE PRUEBAS Y CAMBIOS REALIZADOS ABRIL 2020**

1. **Objetivos principales:**
2. Captura de datos de los componentes físicos de las estrellas de rastreo (Esferas, conjunto y extremo de la herramienta) en dos modos de imagen.
3. Evaluación de estabilidad de datos en los tres modos de imagen.
4. Selección e implementación de filtro para estabilizar datos de coordenadas.
5. **Descripción del proceso:**
6. **5 de abril: Toma de datos para el tipo de video de escala de grises**

**Objetivo:**

Evaluar cómo afecta el tipo de video en escala de grises en la triangulación y en el cálculo de la punta del pointer.

**Procedimiento:**

Se tomaron 1000 datos de cada angulación, en el siguiente orden:

* + - * Pointer inclinado a la derecha a 30°.
      * Pointer inclinado a la izquierda a 30°.
      * Pointer inclinado hacia adelante a 30°.
      * Pointer inclinado hacia atrás a 30°.
      * Pointer centrado.

**Conclusiones:**

* Los datos obtenidos se encuentran en la carpeta “PRUEBASNavarEsferas3D”
* Se recomienda mejorar la sincronización al guardar datos esferas/punta.
* Para el centro, se encuentra que la desviación estándar es de *0.120 mm* con un error máximo de *0.8 mm* para las esferas, y para la punta es de *0.197 mm* de desviación estándar y error máximo de *1.76 mm*.
* Para el pointer inclinado, el error cambia de manera brusca reflejando una desviación estándar de *3.69 mm* y error máximo de *8.83 mm.*
* Se recomienda evaluar por qué existe esta degradación cuando el pointer se inclina cambiando el modo de imagen para observar si cambia la dispersión.

1. **6 de abril: Modificación tipo de video.**

**Objetivo:**

Implementar los tipos de video “Segment” y “Object” para posterior evaluación.

**Procedimiento:**

* Se hace una copia del repositorio de Github y se crea otra rama para aislar los otros tipos de imagen llamada OtherVideoTypes.
* Identifica marcadores detectados en la escena, toma cada una de sus propiedades tales como área y posición en pixeles almacena la información recolectada en matrices pasadas por punteros en el método GetObjects2().

**Conclusiones:**

* Se realiza de esta manera para no afectar el proceso de tracking y toma de datos de Navar original.

1. **11 de Abril: Toma de datos para 3 tipos de video**

**Objetivo:**

Evaluar sí la fuente de error es el tipo de video usado.

**Procedimiento:**

* 1. Se ubica el pointer frente a las cámaras y se capturan 1000 datos por cada posición de este, así:
     + Pointer centrado.
     + Pointer inclinado a la izquierda a 30°.
     + Pointer inclinado a la derecha a 30°.
     + Pointer inclinado hacia adelante a 30°.
     + Pointer inclinado hacia atrás a 30°.
  2. Una vez se capturan 1000 datos en una posición y en un tipo de imagen, se cambia al siguiente tipo de imagen.
  3. En total son cerca de treinta mil datos capturados.

**Conclusiones:**

* Se sigue observando que existe un rango tolerable en la precisión de la punta para el eje central, pero una vez se inclina, el salto se sigue observando.
* Se acuerda re tomar datos desde otro punto del código obteniendo la triangulación para observar si existe algún cambio.
* Se acuerda guardar y/o observar el comportamiento de la matriz PARAM debido a que es una matriz de rotación asociada a un sistema de ejes local evaluado para el pointer el cual mantiene su calibración.
* Los datos obtenidos se encuentran en la carpeta “DiferenteTipoVideo”

1. **15 de abril: Toma de datos para 3 tipos de video capturando datos desde otro punto del código y evaluación del cambio de la matriz PARAM del pointer.**

**Objetivo:** Verificar sí la fuente de error es el tipo de video usado tomando datos desde otro punto del código.

**Procedimiento:**

* Se ubica el pointer centrado frente a las cámaras y se capturan 1000 datos.
* Una vez se capturan 1000 datos en un tipo de imagen, posteriormente se cambia al siguiente tipo de imagen.
* En total son cerca de tres mil datos capturados.
* Se guarda la matriz PARAM en cada ejecución.

**Conclusiones:**

* Se puede observar en los cambios de tipo de video, que la dispersión es menor en el modo Object, esto debido a que la latencia de procesamiento es más baja que en otros modos manteniendo la alta precisión.
* Se opta por cambiar para el rastreo de objetos al modo Objeto ya que además que es el recomendable para el tipo de cámaras que estamos usando, también es bastante útil para sistemas que desean obtener datos en 3D.
* Se observa que la matriz PARAM no se ve afectada en la ejecución.
* Se recomienda evaluar la dispersión navegando dentro del plano visto por las cámaras.
* Los datos obtenidos se encuentran en la carpeta “DatosEsferasdifVideo”

1. **16 de Abril: Toma de datos para evaluación de dispersión teniendo en cuenta la profundidad**

**Objetivo:** Evaluar la dispersión navegando dentro del plano visto por las cámaras trabajando con el modo objeto.

**Procedimiento:**

* Se etiquetan en el espacio 9 posiciones para evaluar y se van tomando datos del Pointer centrado.
* Se navega posición por posición sin interrumpir la ejecución dentro de cada una sino pausarla.
* Se obtiene la posición de las cuatro esferas.
* Se toman mil datos por cada posición, en total 9000 datos capturados aproximadamente.

**Conclusiones:**

* El sistema detecta y captura satisfactoriamente el pointer en las nueve posiciones.
* Se observa que hay mayor distorsión de los datos en los bordes, sobre todo en las posiciones dos y tres.
* Se observa que hay una mejora considerable en la precisión de los datos usando el tipo de video Objeto.
* Se recomienda obtener datos en pixeles y en 3D simultáneos para verificar el cálculo correcto del proceso y la triangulación y compararla con el cálculo hecho por el profesor Jaime Meneses.
* Los datos obtenidos se encuentran en la carpeta “DatosPlano”.

1. **18 de Abril: Toma de datos para evaluar 2D y 3D**

**Objetivo:** obtener datos en pixeles y en 3D simultáneos para verificar el cálculo correcto del proceso y la triangulación.

**Procedimiento:**

* Se realiza la calibración de las cámaras.
* Se navega con el pointer centrado en los puntos 4, 5 y 6 del plano:
* Se capturan datos en cada posición de los datos arrojados por las cámaras en pixeles y del calculo hecho en el proceso de triangulación.

**Conclusiones:**

* Se compara la triangulación hecha por Navar y la triangulación manual hecha por el profesor Jaime, y se observa que el proceso actúa de manera muy similar.
* Se recomienda obtener datos con el pointer inclinado en un punto del Phantom.
* Los datos obtenidos se encuentran en la carpeta “TomaDatos2Dy3D”.

1. **19 de abril: Toma de datos para evaluar inclinación en un punto del Phantom.**

**Objetivo:** Observar la distorsión al inclinar el Pointer en un punto central del Phantom.

**Procedimiento:**

* Ubicar el Pointer en el punto 20 del Phantom y en la posición 5 del plano evaluado.
* Tomar mil datos y guardarlos.
* Inclinar el pointer, tomar mil datos y guardarlos.

**Conclusiones:**

* Se observa que aun cambiando el tipo de video a “Object” sigue existiendo aumento de error al inclinarse el pointer.
* Los datos obtenidos se encuentran en la carpeta “Inclinacion1PuntoPhantom”.

1. **21 de Abril: Obtención de par de imágenes por frame**

**Objetivo:** Evaluar el proceso desde la captura del frame hasta el cálculo de la triangulación.

**Procedimiento:**

1. Se crea un método para capturar un frame por cada cámara y guardarlo como archivo .bmp
2. Se ubica el pointer en una posición y se capturan los siguientes datos:

* Dos imágenes de los frames de cada cámara.
* Una captura datos de posición en pixeles de las esferas.
* Una captura de datos calculados de la triangulación de las esferas.

1. Se evalúan dos posiciones:

* Pointer centrado.
* Pointer inclinado.

**Conclusiones:**

* Se observa que sigue existiendo dentro de un rango tolerable de 1 mm con el pointer centrado, error de posición desde la captura del frame.
* Se observa que sigue aumentando el error con el pointer inclinado desde la captura del frame.
* Los datos obtenidos se encuentran en la carpeta “PruebaImagenFrames”.

1. **22 de Abril: Obtención de par de imágenes de calibración**

**Objetivo:** Obtener imágenes en el modulo de calibración en escala de grises.

**Procedimiento:**

* Calibrar las cámaras
* Ubicar el pointer centrado
* Volver a hacer el procedimiento de calibración en el modulo correspondiente pero solo con una captura.

**Conclusiones:**

* Se realiza esta captura para observar el cambio desde la escala de grises que proporciona la captura.
* Los datos obtenidos se encuentran en la carpeta “ImagendeCalibracionGrises”.

1. **22 de Abril: Captura simultánea de imágenes de calibración y tracking con Pointer centrado e inclinado**

**Objetivo:** Capturar imágenes en el módulo de calibración y tracking.

**Procedimiento:**

* Calibrar las cámaras.
* Hacer la captura en dos posiciones del pointer:
  + Centrado.
  + Inclinado.
* En cada captura:
  + Volver a ingresar al módulo de calibración y capturar una imagen.
  + Ingresar al sistema de rastreo y capturar el primer frame en .bmp, los datos en pixeles y los datos de la triangulación.

**Conclusiones:**

* Los datos obtenidos se encuentran en la carpeta “InclinadoCentrado”.

1. **22 de Abril: Obtención de par de imágenes del tablero localizado en el mismo punto del pointer.**

**Objetivo:** Ubicar el tablero en la misma posición del pointer para verificar.

**Procedimiento:**

* Entrar al módulo de calibración.
* Ubicar el tablero en la misma posición donde está el pointer.
* Capturar una imagen del par de cámaras.

**Conclusiones:**

Se recalibra el sistema agregando la ultima imagen para verificar que este espacio esté explorado.

* Los datos obtenidos se encuentran en la carpeta “CapturaTableroEnPointer”.

1. **23 de Abril: Prueba de filtro.**

**Objetivo**: Observar el cambio en la distorsión aplicando el filtro de promedios.

**Procedimiento:**

* Calibrar las cámaras.
* Asignar 16 a la variable “Sample\_limit” para promediar 16 datos antes de enviarlos a triangular.
* Aumentar el tiempo de exposición para que el tablero se vea más contrastado.
* Poner el Pointer frente a las cámaras y capturar datos según la posición del pointer:
  + Pointer Centrado.
  + Pointer inclinado.
* Medir el tiempo en que tarda en tomar mil datos.

**Conclusiones:**

* La disminución del error es notable:
  + Para pointer centrado, el error máximo tanto para la cámara derecha e izquierda es de 0.15px
  + Para el pointer inclinado, el error máximo para la cámara izquierda es de 0.43px mientras que para la cámara derecha es de 0.36px.
* El tiempo que tarda en tomar mil datos, en cualquier posición es de aproximadamente 20 minutos.
* Se recomienda realizar un análisis y revisar la lectura que hace el sistema ya que Matlab asigna 1,1 para el primer elemento de la imagen mientras que opencv usa 0,0.
* Los datos obtenidos se encuentran en la carpeta “pruebaFilter”.