



Visión computacional estereoscópica

Autor:

Ing. Ariel Salassa

Director:

Esp. Ing. Hernán Contigiani (Globant)

*Esta planificación fue realizada en el curso de Gestión de proyectos
entre el 26 de abril de 2022 y el 14 de junio de 2022.*

Índice

1. Descripción técnica-conceptual del proyecto a realizar.	5
2. Identificación y análisis de los interesados	6
3. Propósito del proyecto.	6
4. Alcance del proyecto	6
5. Supuestos del proyecto.	7
6. Requerimientos	7
7. Historias de usuarios (<i>Product backlog</i>).	8
8. Entregables principales del proyecto	9
9. Desglose del trabajo en tareas	9

Registros de cambios

Revisión	Detalles de los cambios realizados	Fecha
0.0	Creación del documento	26/04/2022
1.0	Se completa hasta la sección 5 inclusive	08/05/2022
1.1	Correcciones de la versión 1.0	20/05/2022
2.0	Se completa hasta la sección 9 inclusive	22/05/2022

Acta de constitución del proyecto

Buenos Aires, 26 de abril de 2022

Por medio de la presente se acuerda con el Ing. Ariel Salassa que su Trabajo Final de la Carrera de Especialización en Inteligencia Artificial se titulará “Visión computacional estereoscópica” y consistirá en investigar el estado del arte de la visión computacional realizada con más de una cámara como entrada, de modo que se puedan tomar decisiones en un ambiente 3D simulado, con fecha de inicio 26 de abril de 2022 y fecha de presentación pública diciembre de 2022.

Se adjunta a esta acta la planificación inicial.

Ariel Lutenberg
Director posgrado FIUBA

Esp. Ing. Hernán Contigiani
Globan

Esp. Ing. Hernán Contigiani
Director del Trabajo Final

1. Descripción técnica-conceptual del proyecto a realizar

La visión computacional en pocas palabras consta de cámaras que entregan fotogramas a un sistema que los toma y los procesa teniendo en cuenta una calibración previa.

La idea del proyecto se basa en el uso de al menos dos cámaras conectadas a un mismo sistema de visión que sea capaz de procesar las imágenes de cada una de las cámaras permitiendo estimar atributos de objetos tales como altura, profundidad y volumen.

Actualmente existen cámaras estereoscópicas integradas en un único dispositivo (Figura 1) que realizan dichas tareas, permitiendo agregar funcionalidades como la navegación de robots o asistencia a conductores. Las cámaras estereoscópicas integradas tienen la ventaja de que el mismo hardware y firmware realiza la inferencia, pero tienen la desventaja de que la distancia entre cámaras es fija y esto no permite tomar espectros 3D en infraestructuras preexistentes.

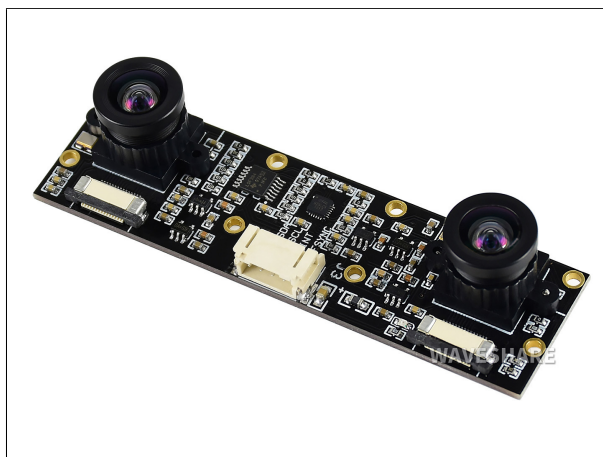


Figura 1. Cámara estereoscópica.

Tener un sistema de cámaras conectado a un servidor puede otorgar mucha más precisión como así también, mejor alcance de la solución. Incluso se podría evaluar la posibilidad de hacer visión computacional estereoscópica con cámaras que no están fijas.

Las partes que conforman un sistema de visión computacional se pueden observar en el siguiente diagrama de la Figura 2.

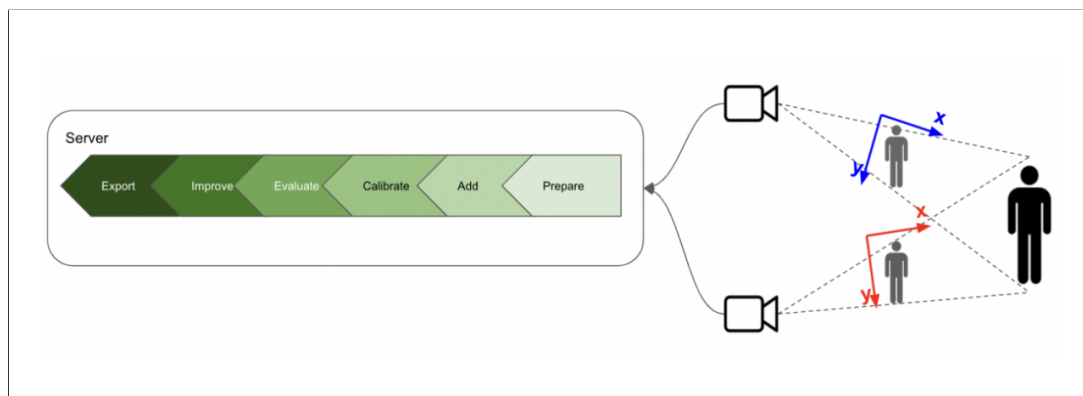


Figura 2. Partes de un sistema de visión computacional.

2. Identificación y análisis de los interesados

Rol	Nombre y Apellido	Organización	Puesto
Responsable	Ing. Ariel Salassa	Autónomo	Sr. Software Engineer Alumno
Cliente	Ing. Juan Pablo Pizarro	Globant	Software Development Specialist
Orientador	Esp. Ing. Hernán Contigiani	Globant	Sr. Software Engineer Director Trabajo final
Usuario final	Desarrolladores de Software	-	Software developers

Cuadro 1. Identificación de los interesados.

- Responsable: Ing. Ariel Salassa, es la persona que desarrollará el proyecto.
- Cliente: El Ing. Juan Pablo Pizarro, es líder y referente técnico del equipo de desarrollo del departamento de IOT de Globant. Él es el principal interesado en que su departamento cuente con alguna solución de visión estereoscópica. Por sus altos grados de conocimientos técnicos también puede ser capaz de orientar en el desarrollo.
- Orientador: Hernán Contigiani es el director del presente proyecto y líder técnico en Globant. Su función será orientar al responsable a lo largo de la realización del proyecto.
- Desarrolladores de Software: son los usuarios finales que podrán hacer uso del sistema para enriquecer sus aplicaciones.

3. Propósito del proyecto

El propósito del proyecto es investigar el estado del arte de la visión computacional con más de una cámara como entrada de video de modo de que se puedan tomar decisiones en un ambiente 3D simulado. Con ello se pretende llegar a un sistema de visión estereoscópica más versátil que aquellos integrados en un dispositivo.

4. Alcance del proyecto

El proyecto comprenderá las siguientes etapas:

- Planificación de tareas.
- Formación en visión computacional.
- Investigación de sistemas utilizados en cámaras estereoscópicas.
- Formación en protocolos de comunicación utilizados para *straming*.
- Configuración y calibración de cámaras.
- Comunicación remota entre las cámaras y el servidor.

- Preparación de los datos.
- Extracción de características.
- Prototipado conceptual funcional.
- Reporte de resultados.

El presente proyecto no incluye:

- Desarrollo fino de redes neuronales que permitan detectar y clasificar objetos.

5. Supuestos del proyecto

Para el desarrollo del presente proyecto se supone que:

- El responsable dispondrá de suficiente cantidad de tiempo para resolver los problemas que se presenten en el desarrollo del proyecto.
- El responsable tendrá a su disposición a su director y/o colaboradores cuando sea pertinente.
- Se puede alcanzar un prototipo del sistema utilizando *webcams*.
- Se dispondrá de suficiente potencia de cómputo para trabajar con *streaming* de imágenes.
- Es posible realizar una composición de dos o más imágenes en una única imagen de salida que permita estimar profundidad y volumen de los objetos presentes en la imagen.
- Se puede aplicar el protocolo RTMP (*Real Time Messaging Protocol*) o RTSP (*Real Time Streaming Protocol*) para la adquisición remota de las imágenes.
- Se puede integrar al sistema una red neuronal pre entrenada para detectar objetos.

6. Requerimientos

1. Requerimientos de documentación

- 1.1. El trabajo debe ser continuamente documentado y se presentarán informes mensuales de avance al director.
- 1.2. Los informes de avance pueden ser presentados como código correctamente documentado con los resultados correspondientes.
- 1.3. El código final será de tipo abierto y deberá quedar ordenado en un repositorio público.

2. Requerimientos de lenguajes y frameworks

- 2.1. La librería para el tratamiento de las imágenes será *OpenCV*.
- 2.2. El código para el desarrollo deberá ser escrito en Python. Si se requiriese una optimización de hardware, entonces deberá utilizarse C++.

2.3. Se deberá priorizar el uso de librerías standard (numpy, pandas, matplotlib, seaborn, etcétera).

3. Requerimientos funcionales

3.1. El sistema de visión debe estar compuesto por al menos dos cámaras.

3.2. Cada cámara debe estar correctamente calibrada.

3.3. El sistema de visión debe poder comandarse de manera remota.

3.4. La transmisión de imágenes desde el sistema de visión hacia el resto del sistema de clasificación y evaluación debe hacerse usando protocolos de transmisión en tiempo real.

3.5. El sistema debe ser capaz de detectar objetos y determinar profundidad, volumen y/o distancia de los mismos.

3.6. El error del sistema no deberá ser superior al 15

4. Requerimientos de testing y evaluación

4.1. Los resultados del sistema deberán ser corroborados previamente en un entorno local y con objetos de dimensiones conocidas.

7. Historias de usuarios (*Product backlog*)

La medida del trabajo a efectuar para cumplir con cada una de las historias de usuarios estará dada por *story points*. Para ponderar los esfuerzos se utilizará la serie de Fibonacci con valores: 0, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, etc. Cuando el esfuerzo se considere alto los *story points* tomarán valores entre 0 y 3 inclusive. Cuando se considere medio, entre 5 y 13 inclusive. Cuando el esfuerzo se considere alto el valor de los *story points* será igual o mayor a 21.

- Como desarrollador quiero un prototipo de visión estereoscópica que pueda extraer características como distancia, profundidad y volumen de objetos.

Dificultad: Alta (34) – Complejidad: Alta (34) – Incertidumbre: Alta (34)

Story Points: $34 + 34 + 34 = 102 \rightarrow 89$

- Como cliente quiero tener un sistema de visión realizado con tecnologías standards y *open source* para poder mantenerlo a futuro con el menor esfuerzo posible.

Dificultad: Media (8) – Complejidad: Media (8) – Incertidumbre: Baja (1)

Story Points: $8 + 8 + 1 = 17 \rightarrow 21$

- Como desarrollador quiero poder acceder remotamente al sistema de visión.

Dificultad: Alta (21) – Complejidad: Alta (21) – Incertidumbre: Alta (21)

Story Points: $21 + 21 + 21 = 63 \rightarrow 55$

- Como cliente quiero tener el prototipo de un sistema de visión que permita futuras iteraciones.

Dificultad: Media (8) – Complejidad: Media (8) – Incertidumbre: Alta (21)

Story Points: $8 + 8 + 21 = 37 \rightarrow 34$

- Como cliente quiero recibir la documentación del trabajo realizado para que pueda servir como base de futuros desarrollos de la empresa.

Dificultad: Media (13) – Complejidad: Baja (2) – Incertidumbre: Baja (1)

Story Points: $13 + 2 + 1 = 16 \rightarrow 13$

- Como cliente quiero tener una presentación resumida del trabajo para mostrar sus resultados y su potencial a todos los interesados.

Dificultad: Baja (3) – Complejidad: Baja (2) – Incertidumbre: Baja (2)

Story Points: $3 + 2 + 2 = 7 \rightarrow 8$

8. Entregables principales del proyecto

Los entregables del proyecto son:

- Informe final.
- Presentación final.
- Diagrama de funcionamiento.
- Prototipo conceptual funcional.
- Repositorio de código del sistema.
- Informe de avance.

9. Desglose del trabajo en tareas

1. Planificación. (60 hs)

- 1.1. Estudio de necesidades. (9 hs)
- 1.2. Análisis de factibilidad. (3 hs)
- 1.3. Definición de requerimientos. (9 hs)
- 1.4. Confección de documento de planificación. (39 hs)

2. Investigación y capacitación. (108 hs)

- 2.1. Capacitación en conceptos básicos de visión estereoscópica y sus aplicaciones. (9 hs)
- 2.2. Capacitación en geometría de la formación de imágenes. (6 hs)
- 2.3. Capacitación en geometría epipolar. (6 hs)
- 2.4. Capacitación en distorsión de lentes. (6 hs)
- 2.5. Capacitación en calibración de cámaras. (6 hs)
- 2.6. Capacitación en *OpenCV*. (21 hs)
- 2.7. Capacitación en protocolos de *streaming* en tiempo real. (15 hs)
- 2.8. Elaboración de códigos de ejemplos básicos. (45 hs)

3. *Setup* de cámaras. (45 hs)

- 3.1. Conexión de cámaras y acceso al *stream* de video mediante ordenador. (15 hs)
- 3.2. Calibración de cámaras individuales. (15 hs)
- 3.3. Montaje de cámaras y calibración conjunta. (15 hs)
- 4. Construcción del *pipeline* de preparación de datos. (51 hs)
 - 4.1. Suavizado y ajuste de tamaño. (15 hs)
 - 4.2. Filtro de distorsión de lentes. (21 hs)
 - 4.3. Evaluación y selección de FPS (*frames per second*) del sistema. (15 hs)
- 5. Evaluación estéreo. (105 hs)
 - 5.1. Detección de objetos de cada imagen. (15 hs)
 - 5.2. Implementación de *template matching*. (15 hs)
 - 5.3. Implementación de geometría epipolar. (15 hs)
 - 5.4. Implementación de disparidad estéreo. (15 hs)
 - 5.5. Cálculo de distancias y profundidades. (15 hs)
 - 5.6. Estimación volumétrica. (15hs)
 - 5.7. Prototipado y preparación de resultados de salida. (15 hs)
- 6. Pruebas de validación. (63 hs)
 - 6.1. Pruebas locales en tiempo real. (21 hs)
 - 6.2. Pruebas remotas en tiempo real. (21 hs)
 - 6.3. Ajustes finales. (21 hs)
- 7. Documentación y presentación final. (171 hs)
 - 7.1. Elaborar informe de avance proyecto. (60 hs)
 - 7.2. Elaborar informe final del proyecto. (90 hs)
 - 7.3. Preparación de presentación final. (21 hs)

Cantidad total de horas: (603 hs)