Algoritmo de paralelización para la estimación en tiempo real del ángulo de guiñada de un AUV

Claudio Paz, Gastón Araguás, Gonzalo Perez Paina, Hugo Toloza

Argencon 2014
Segundo Congreso Bienal de IEEE Argentina
San Carlos de Bariloche, Rio Negro, Argentina
11-13 de junio

Universidad Tecnológica Nacional
Facultad Regional Córdoba

Centro de Investigación
en Informática para la Ingeniería

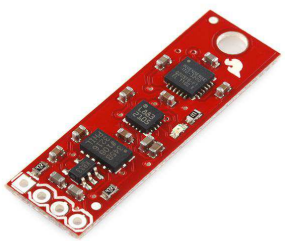
Outline

Introducción y objetivos

Unmanned Aerial Vehicles (UAV)

- Ventajas: baratos, facil construcción y mantenimiento, livianos y fácil de controlar.
- Desventajas: carga útil y recursos de cómputo limitados.

Algunas aplicaciones como inspección o navegación en interiores requieren vuelo estacionario.



Sensores inerciales

Cámaras

Sonar

Magnetómetros

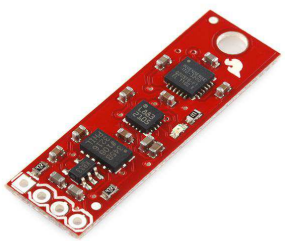
GPS

Introducción y objetivos

Unmanned Aerial Vehicles (UAV)

- Ventajas: baratos, facil construcción y mantenimiento, livianos y fácil de controlar.
- Desventajas: carga útil y recursos de cómputo limitados.

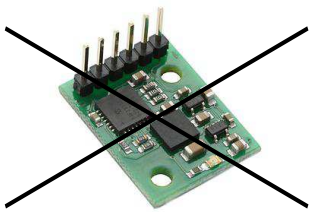
Algunas aplicaciones como inspección o navegación en interiores requieren vuelo estacionario.



Sensores inerciales

Cámaras

Sonar



Magnetómetros

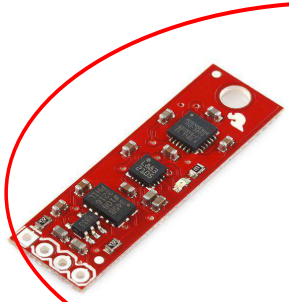
GPS

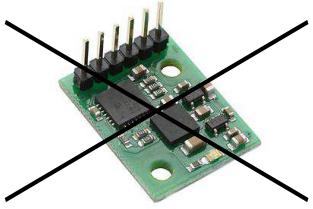
Introducción y objetivos

Unmanned Aerial Vehicles (UAV)

- Ventajas: baratos, facil construcción y mantenimiento, livianos y fácil de controlar.
- Desventajas: carga útil y recursos de cómputo limitados.

Algunas aplicaciones como inspección o navegación en interiores requieren vuelo estacionario.



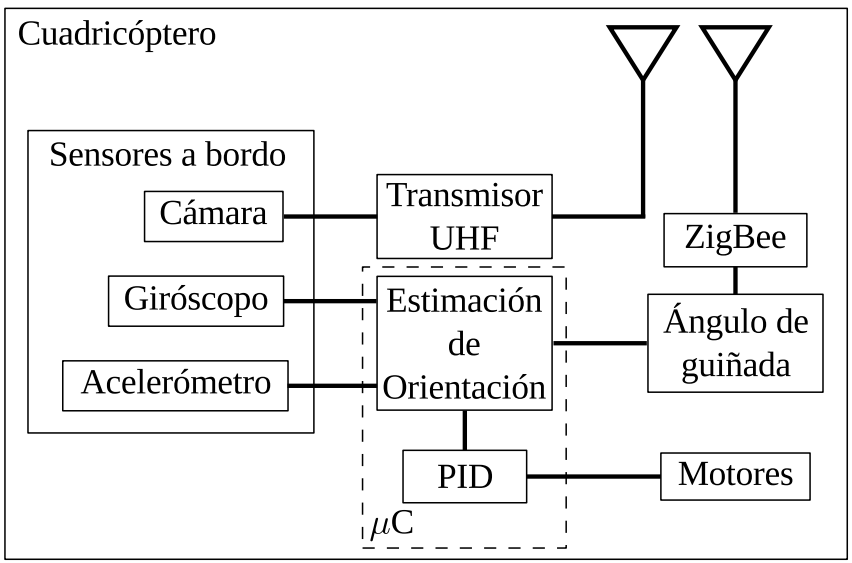



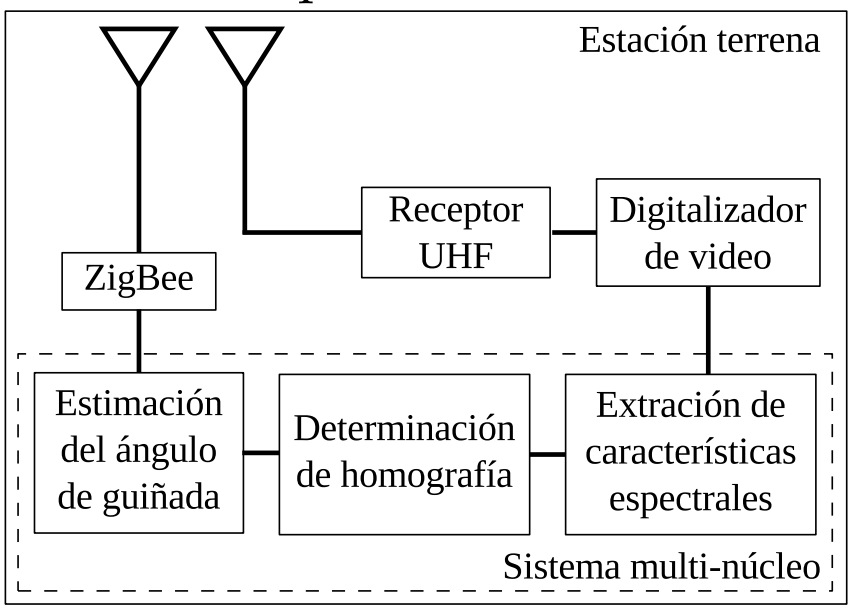

Sensores inerciales Cámaras Sonar Magnetómetros GPS

Objetivo: Estimar el ángulo de guiñada (yaw) mediante fusión camara-IMU en tiempo real.

Diagrama en bloques

Sistema de estimación de orientación en tiempo real.





Estimación visual del ángulo de guiñada

Según el modelo pinhole

$$s\boldsymbol{m} = \boldsymbol{K}[\boldsymbol{R}|\boldsymbol{t}] \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = \boldsymbol{K} \begin{bmatrix} \boldsymbol{r}_1 & \boldsymbol{r}_2 & \boldsymbol{t} \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = \boldsymbol{H}\boldsymbol{M}$$

Considerando una cámara en movimiento, en el instante t_a

$$s_a\boldsymbol{m}_a = \boldsymbol{H}_{wa}\boldsymbol{M}$$

en el instante t_b

$$s_b\boldsymbol{m}_b = \boldsymbol{H}_{wb}\boldsymbol{M}$$

Estimación visual del ángulo de guiñada

Según el modelo pinhole

$$s\boldsymbol{m} = \boldsymbol{K}[\boldsymbol{R}|\boldsymbol{t}] \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = \boldsymbol{K} \begin{bmatrix} \boldsymbol{r}_1 & \boldsymbol{r}_2 & \boldsymbol{t} \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = \boldsymbol{H}\boldsymbol{M}$$

Considerando una cámara en movimiento, en el instante t_a

$$s_a\boldsymbol{m}_a = \boldsymbol{H}_{wa}\boldsymbol{M}$$

en el instante t_b

$$s_b\boldsymbol{m}_b = \boldsymbol{H}_{wb}\boldsymbol{M}$$

si el movimiento es suave, se puede asumir que $s_a \approx s_b$

$$\boldsymbol{m}_a \approx \boldsymbol{H}_{ba}\boldsymbol{m}_b$$

con $\boldsymbol{H}_{ba} = \boldsymbol{H}_{wa}(\boldsymbol{H}_{wb})^{-1}$.

Estimación visual del ángulo de guiñada (continuación)

Despreciando rolido y cabeceo:

$$\boldsymbol{m}_a = \boldsymbol{H}_{ba}\boldsymbol{m}_b \approx \begin{bmatrix} \boldsymbol{R}_z & \boldsymbol{t} \\ \boldsymbol{0} & 1 \end{bmatrix} \boldsymbol{m}_b$$

donde $\boldsymbol{R}_z \in SO(2)$ es la matriz de rotación alrededor del eje focal y $\boldsymbol{t} \in \mathbb{R}^2$ es vector de traslación del CCS entre los instantes t_a y t_b.

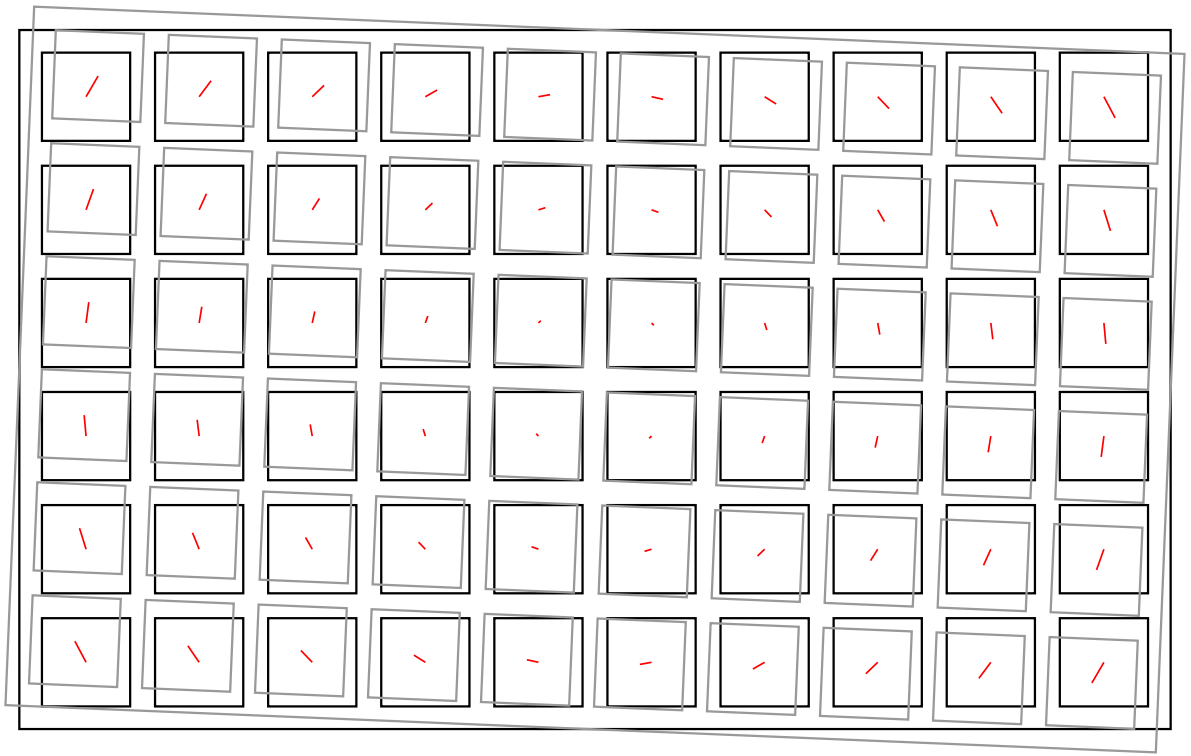
$$\boldsymbol{R}_z = \begin{bmatrix} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{bmatrix}$$

con ψ el ángulo de guiñada.

Características espectrales

Cálculo de correspondencias

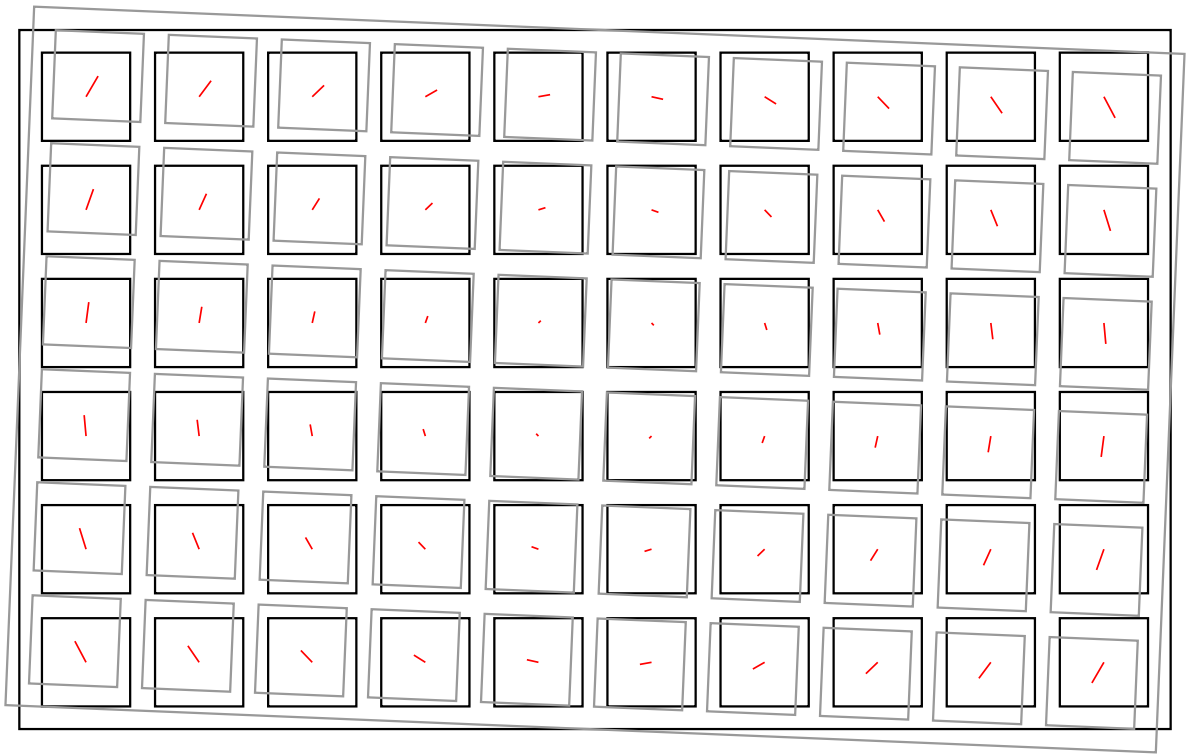
Para determinar la homografía $\boldsymbol{H}_{ba}$ se debe encontrar el conjunto de correspondencias $\{\boldsymbol{m}_{a_i} \leftrightarrow \boldsymbol{m}_{b_i}\}$.



Características espectrales

Cálculo de correspondencias

Para determinar la homografía $\boldsymbol{H}_{ba}$ se debe encontrar el conjunto de correspondencias $\{\boldsymbol{m}_{a_i} \leftrightarrow \boldsymbol{m}_{b_i}\}$.

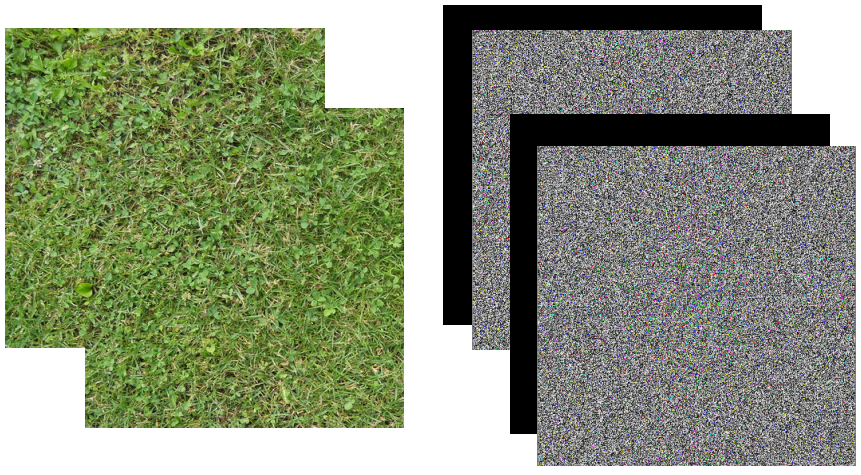


Características espectrales (continuación)

Correlación de fase (PCM)

$$i_a(x, y) = i_b(x + u, y + v)$$

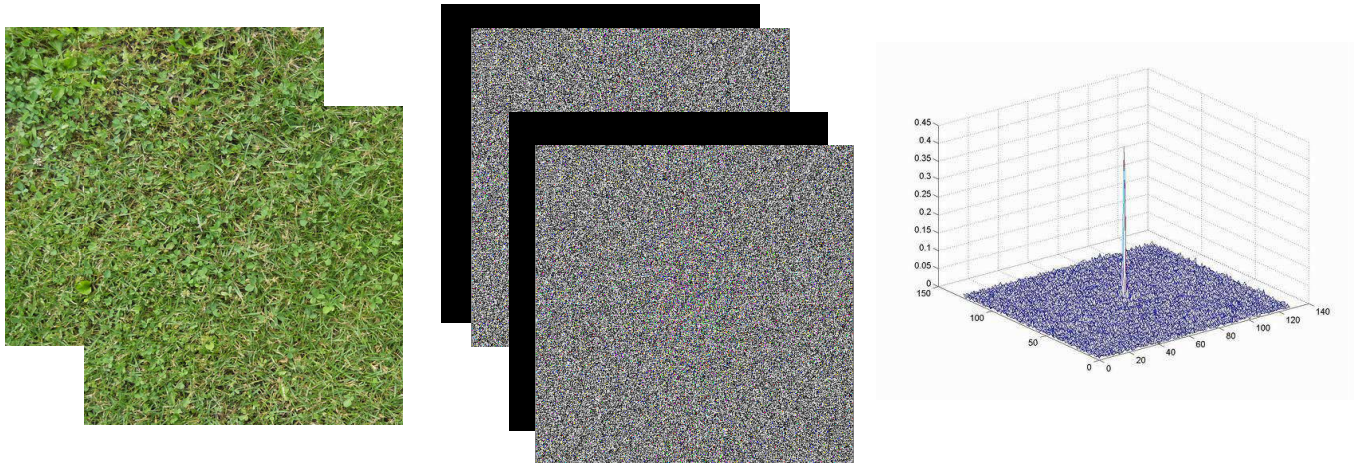
Características espectrales (continuación)



Correlación de fase (PCM)

$$i_a(x,y) = i_b(x+u, y+v) \quad \rightarrow \quad I_a(\omega_x, \omega_y) = e^{j(u\omega_x + v\omega_y)} I_b(\omega_x, \omega_y)$$

Características espectrales (continuación)



Correlación de fase (PCM)

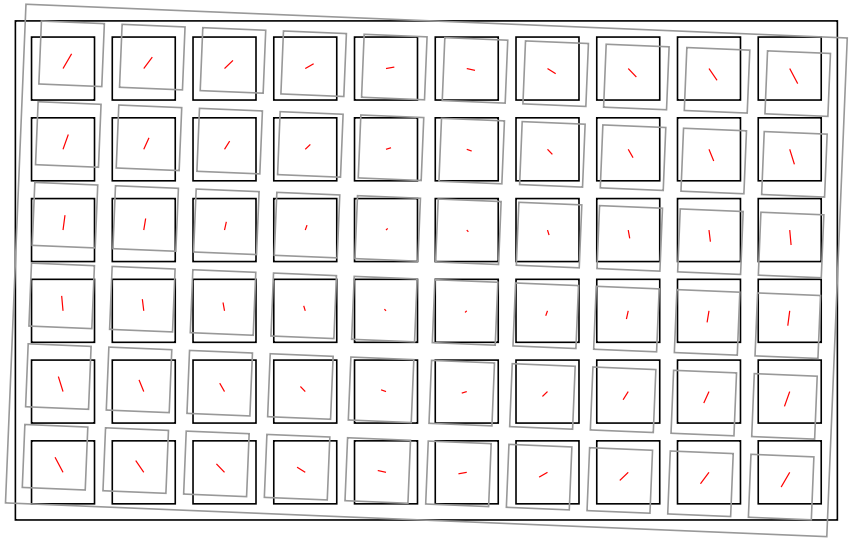
$$i_a(x,y) = i_b(x+u, y+v) \quad \rightarrow \quad I_a(\omega_x, \omega_y) = e^{j(u\omega_x + v\omega_y)} I_b(\omega_x, \omega_y)$$

$$\mathcal{C}(I_a, I_b) = \frac{I_a(\omega_x, \omega_y) I_b^*(\omega_x, \omega_y)}{|I_a(\omega_x, \omega_y)||I_b^*(\omega_x, \omega_y)|} = e^{j(u\omega_x + v\omega_y)}$$

Características espectrales (continuación)

Definiendo los parches y estimando su desplazamiento se construye el conjunto de correspondencias

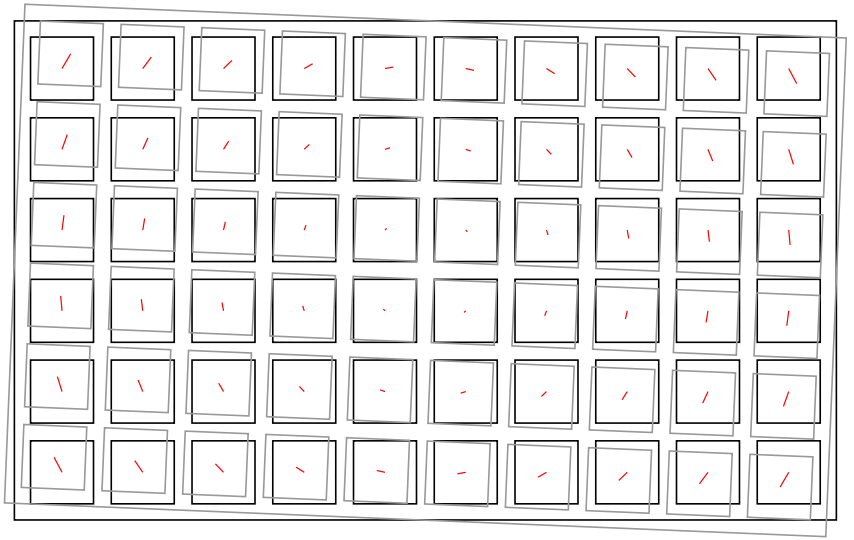
$$\{\boldsymbol{m}_{a_i} \leftrightarrow \boldsymbol{m}_{a_i} + \Delta\boldsymbol{d}_i = \boldsymbol{m}_{b_i}\}$$



Características espectrales (continuación)

Definiendo los parches y estimando su desplazamiento se construye el conjunto de correspondencias

$$\{\boldsymbol{m}_{a_i} \leftrightarrow \boldsymbol{m}_{a_i} + \Delta\boldsymbol{d}_i = \boldsymbol{m}_{b_i}\}$$



---

function ESTIMACION_GUIÑADA(i_t, i_{t-1})
  Obtener parches $p_{i\,t}$ y $p_{i\,t-1}$ a partir de i_t y i_{t-1}
  for all $\{p_{i\,t}, p_{i\,t-1}\}$ **do**
    $\Delta\boldsymbol{d}_i \leftarrow$ DESPLAZAMIENTO($p_{i\,t}, p_{i\,t-1}$)
    $\boldsymbol{m}_{i\,t} \leftarrow \boldsymbol{m}_{i\,t-1} + \Delta\boldsymbol{d}_i$
  end for
  $\psi \leftarrow \boldsymbol{R}_z \leftarrow \boldsymbol{H} \leftarrow$ HOMOGRAFÍA($\boldsymbol{m}_{i\,t}, \boldsymbol{m}_{i\,t-1}$)
  return ψ
end function

Implementación en arquitectura multi-nodo

Consideraciones

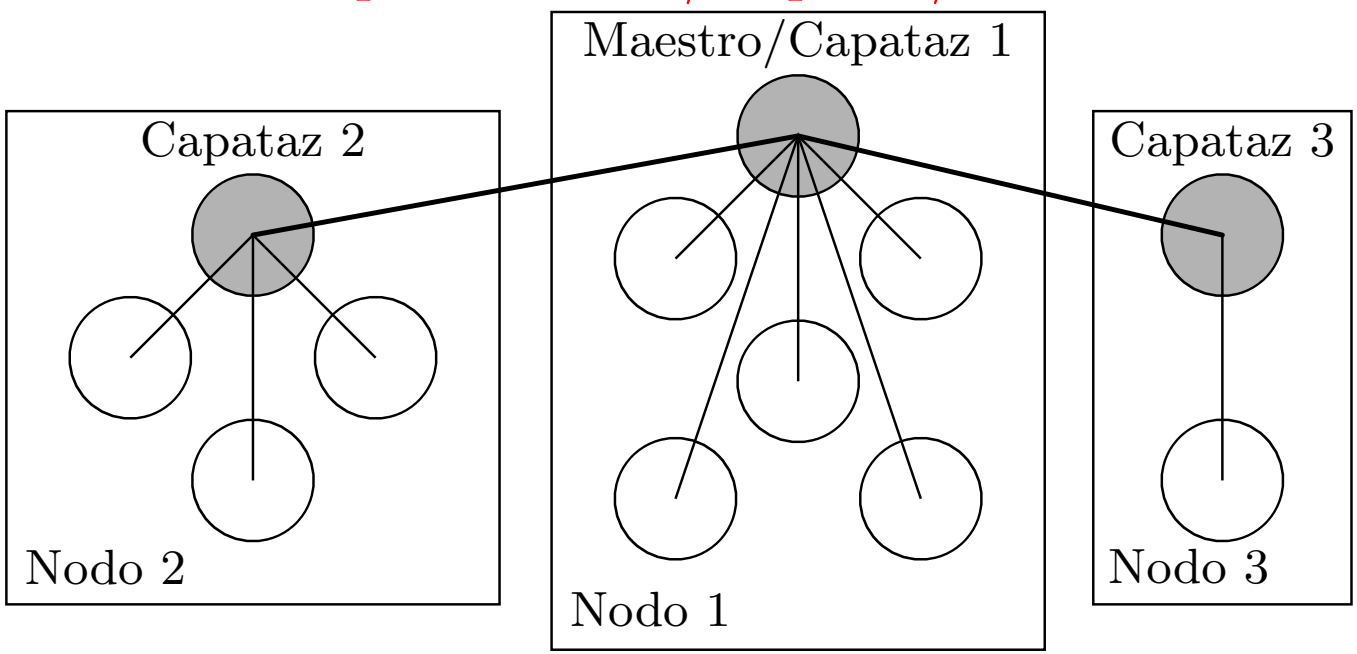
- División en parches → Particionado de dominio
- Computadoras multi-nucleos → Memoria compartida sin latencia de red
- Implementación paralela → MPI (www.mpich.org)

Implementación en arquitectura multi-nodo

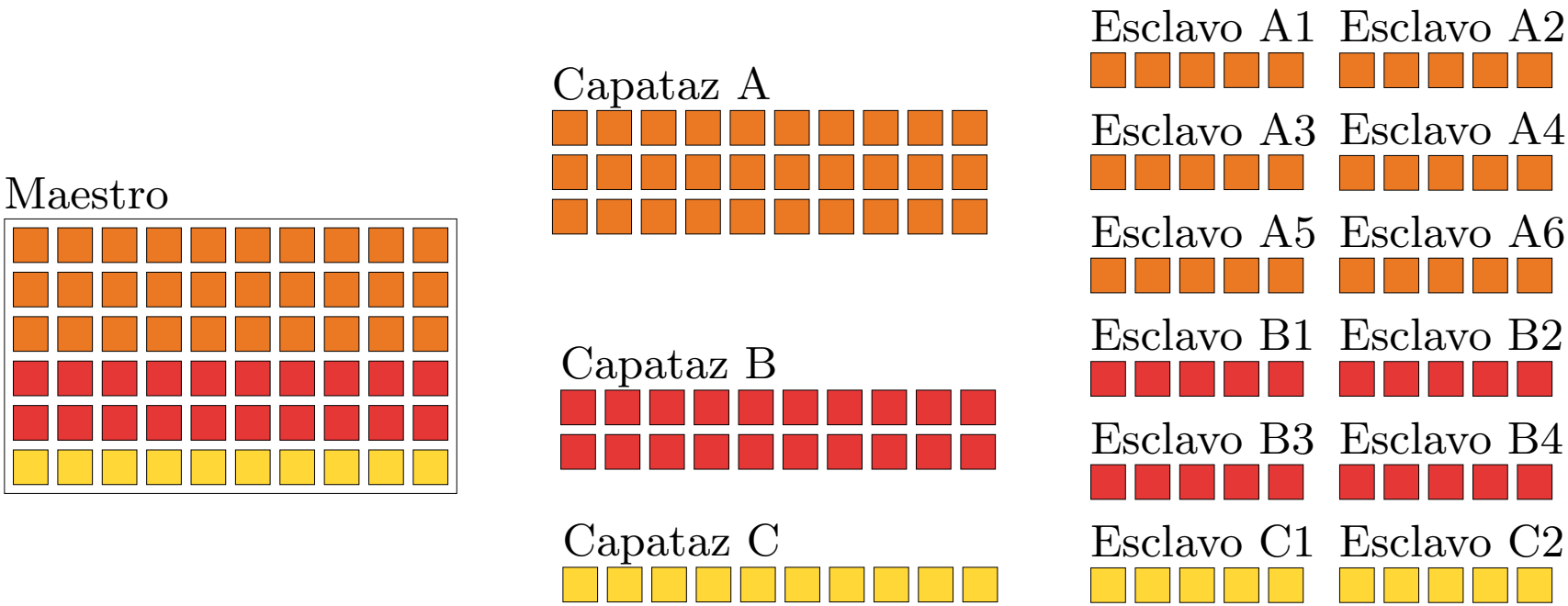
Consideraciones

- División en parches → Particionado de dominio
- Computadoras multi-nucleos → Memoria compartida sin latencia de red
- Implementación paralela → MPI (www.mpich.org)

Jerarquía Maestro/Capataz/Esclavo



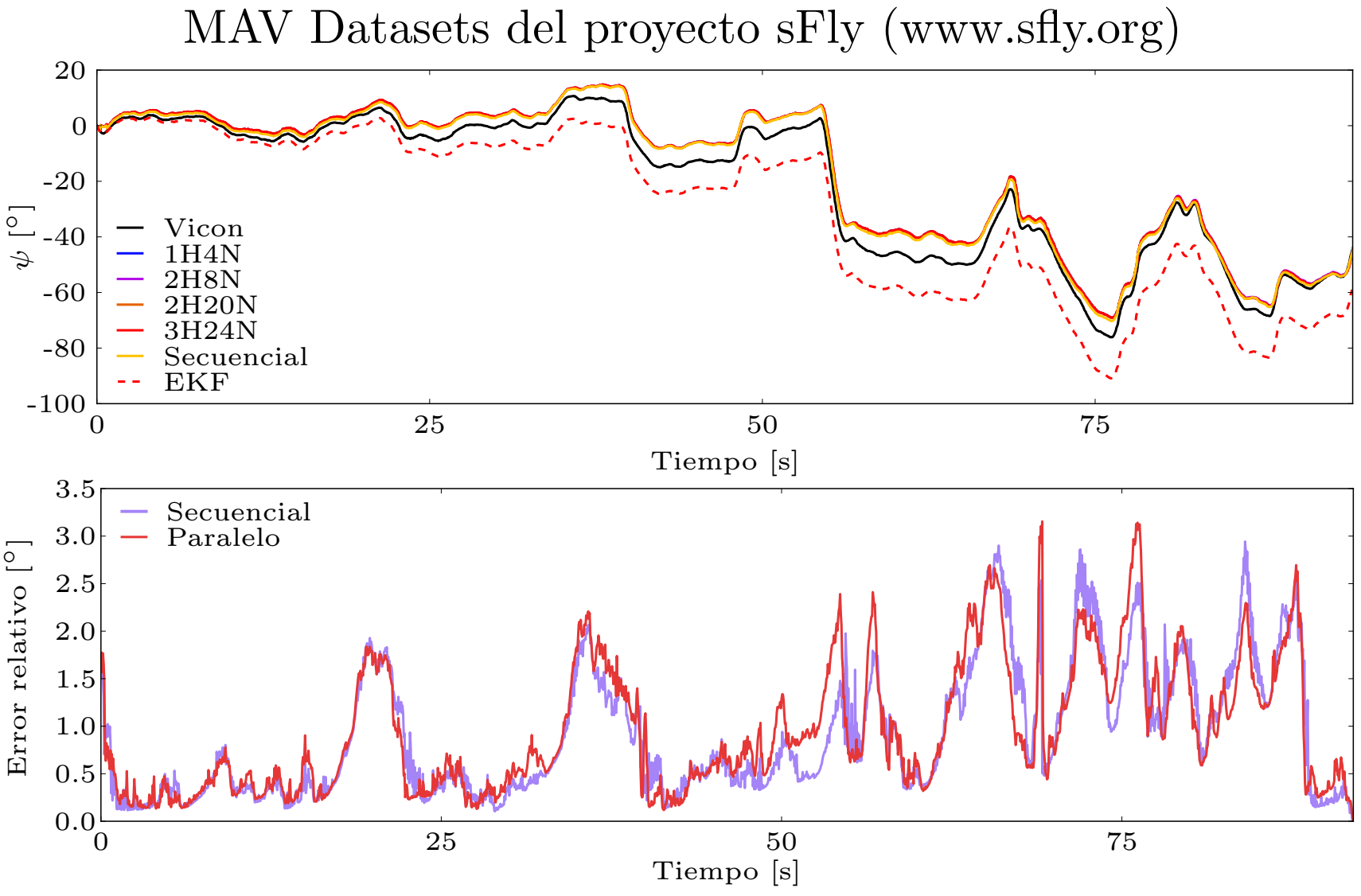
Implementación en arquitectura multi-nodo (cont.)



Implementación en arquitectura multi-nodo (cont.)

function ESTIMACION_GUIÑADA_PARALELA(i_t, i_{t-1})
  if Procesador == Maestro **then**
    Obtener bloques $b_{j\,t}$ y $b_{j\,t-1}$ a partir de I_t y I_{t-1}
    Enviar bloques a los j capataces
  end if
  if Procesador == Capataz **then**
    Recibir bloques del maestro
    Obtener parches $p_{i\,t}$ y $p_{i\,t-1}$ a partir de $b_{j\,t}$ y $b_{j\,t-1}$
    Enviar parches a los i esclavos
  end if
  if Procesador == Esclavo **then**
    $\Delta \boldsymbol{d}_i \leftarrow$ DESPLAZAMIENTO($p_{i\,t}, p_{i\,t-1}$)
  end if
  GATHER($\Delta \boldsymbol{d}_i$)
  if Procesador == Maestro **then**
    $\boldsymbol{m}_{i\,t} \leftarrow \boldsymbol{m}_{i\,t-1} + \Delta \boldsymbol{d}_i$
    $\psi \leftarrow \boldsymbol{R}_z \leftarrow \boldsymbol{H} \leftarrow$ HOMOGRAFÍA($\boldsymbol{m}_{i\,t}, \boldsymbol{m}_{i\,t-1}$)
    return ψ
  end if
end function

Resultados

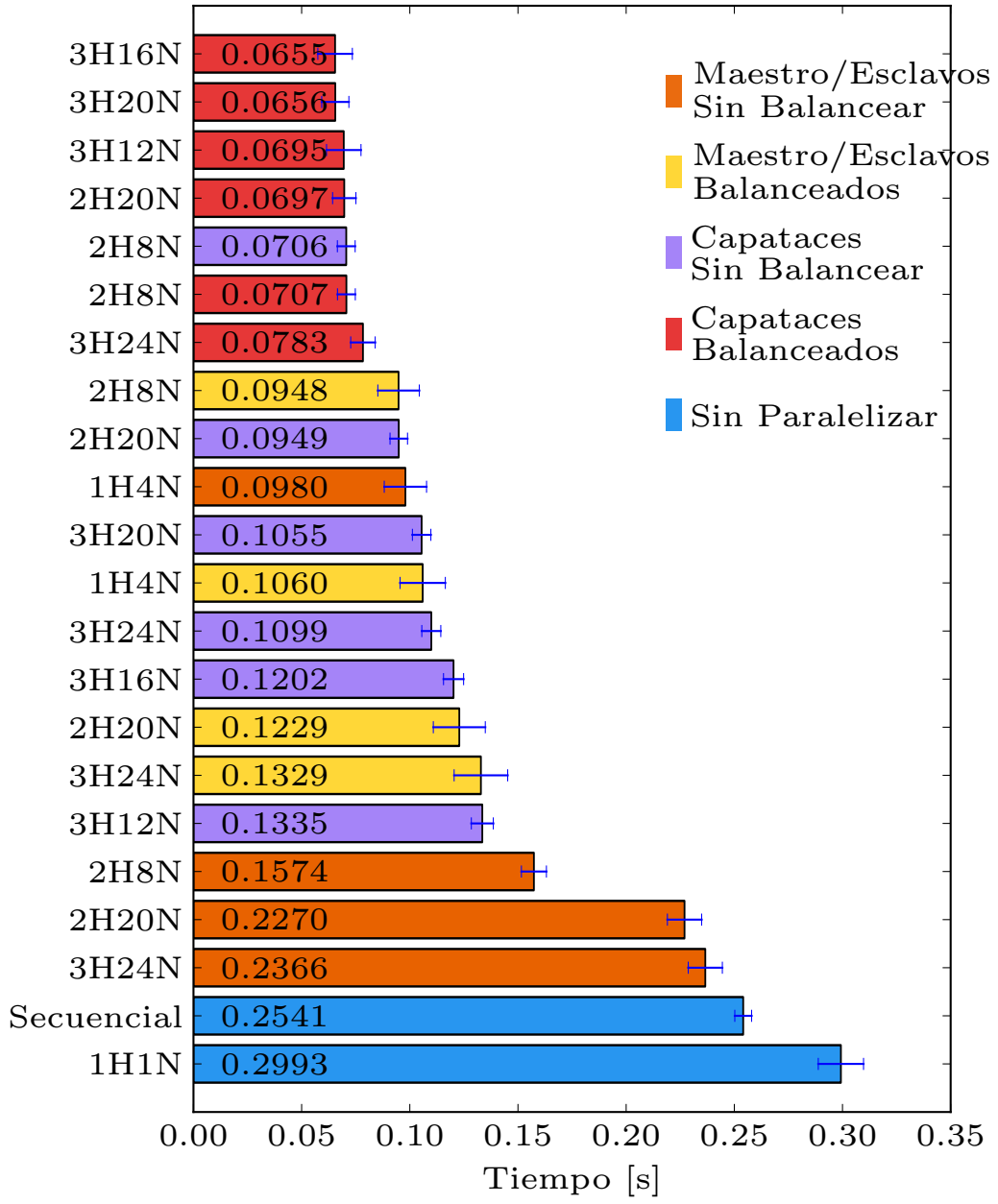

MAV Datasets del proyecto sFly (www.sfly.org)
ψ [°]
Vicon
1H4N
2H8N
2H20N
3H24N
Secuencial
EKF
Tiempo [s]
Error relativo [°]
Secuencial
Paralelo
Tiempo [s]

Resultados (continuación)

Speedup algorítmico

$$s_N = \frac{t_1}{t_N}$$

$$s_{16} = 3{,}88$$



Conclusiones y trabajo futuro

Conclusiones

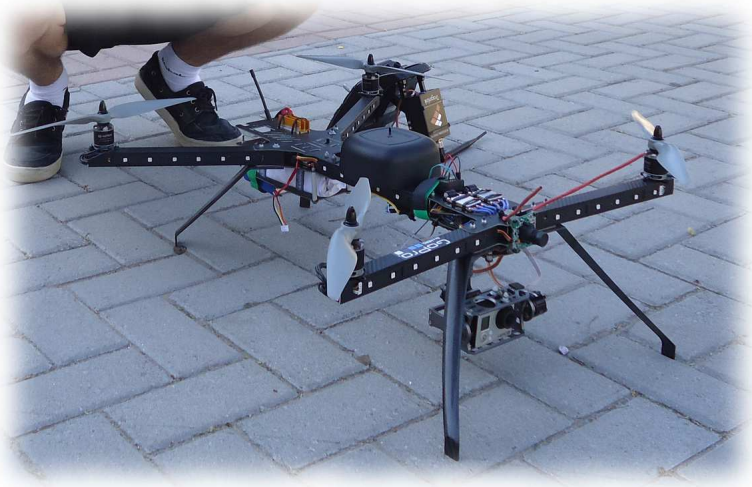
- Particionado de dominio para aprovechar equipos heterogeneos.
- Maestro/Capataz/Esclavo oculta la latencia de red.
- La escalabilidad del algoritmo está sujeta a la cantidad de parches.

Trabajo futuro

- Optimización de función `cv::phaseCorrelate()`
- Implementación para GPU (Jetson TK1: Cortex A15 quad-core, con GPU onboard.)

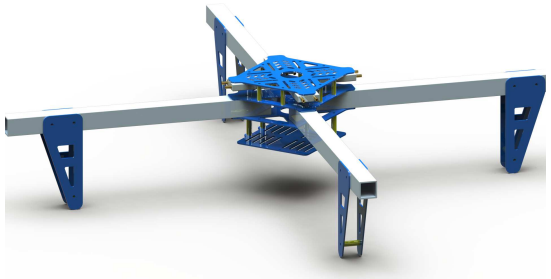
Algoritmo de paralelización para la estimación en tiempo real del ángulo de guiñada de un AUV

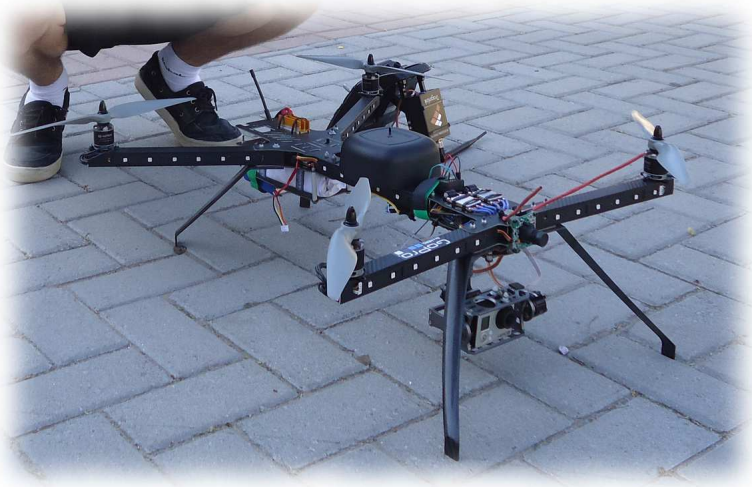
Claudio Paz
Gastón Araguás
{cpaz,garaguas}@scdt.frc.utn.edu.ar



Algoritmo de paralelización para la estimación en tiempo real del ángulo de guiñada de un AUV

Claudio Paz
Gastón Araguás
{cpaz,garaguas}@scdt.frc.utn.edu.ar





Gracias.