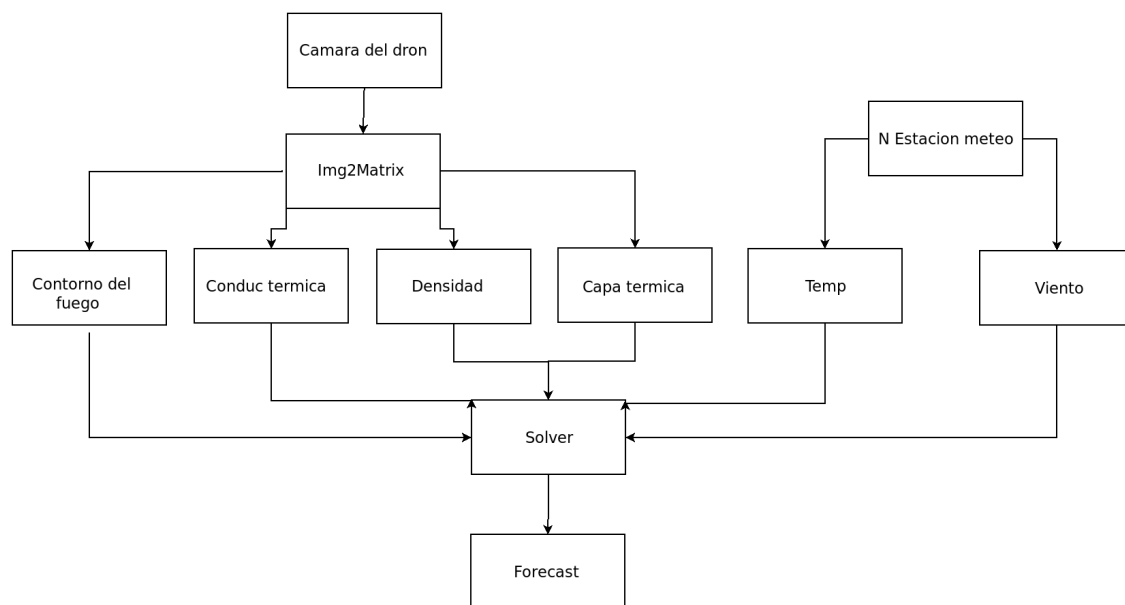


El-posta

February 21, 2020

1 Ngen: modelando el incendio de los bosques



```
[1]: import numpy as np
from PIL import Image
import matplotlib.pyplot as plt

from ngen.model import Solver
from ngen.preprocessing import img2matrix
```

1.1 Mapeo de imagenes

Las imagenes obtenidas con el dron las mapeamos a las distintas matrices que necesita el modelo como input: coordenadas del fuego y mapeo de los materiales del bosque.





```
[2]: # Get 4 matrix:
# - Fire position
# - Thermal conductivity of each pixel (K)
# - Heat capacity of each pixel (Cp)
```

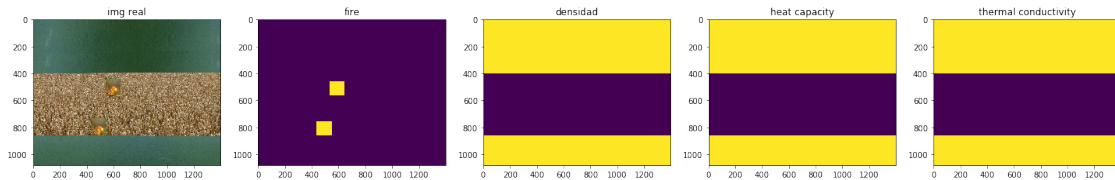
```
# - Density of each pixel (Rho)

img_path = 'imgs/test1.jpg'
pixel_size = 2.7 # in meters
fire, Rho, Cp, K = img2matrix(img_path, pixel_size)
```

```
[3]: img_real = np.array(Image.open(img_path))
titles = ['img real', 'fire', 'densidad', 'heat capacity', 'thermal_
↳conductivity']

fig, axes = plt.subplots(1, 5, figsize=(25, 6))
for ax, img, title in zip(axes, [img_real, fire, Rho, Cp, K], titles):
    ax.imshow(img)
    ax.set_title(title)

plt.show()
```

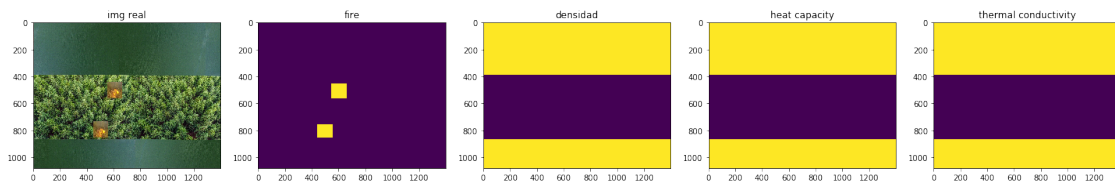


```
[4]: img_path = 'imgs/test2.jpg'
pixel_size = 2.7 # in meters
fire, Rho, Cp, K = img2matrix(img_path, pixel_size)

img_real = np.array(Image.open(img_path))
titles = ['img real', 'fire', 'densidad', 'heat capacity', 'thermal_
↳conductivity']

fig, axes = plt.subplots(1, 5, figsize=(25, 6))
for ax, img, title in zip(axes, [img_real, fire, Rho, Cp, K], titles):
    ax.imshow(img)
    ax.set_title(title)

plt.show()
```



2 Modelado

Partiendo de la ecuacion del calor de Fourier (en dos dimensiones inicialmente) y considerando materiales no isotropicos que varian en funcion de la posicion y de la temperatura y agregando un termino para modelar la influencia del viento obtenemos:

$$\frac{\partial T}{\partial t} = \rho c_p \left[\frac{\partial}{\partial x} \left(k \frac{\partial T}{\partial x} \right) + \frac{\partial}{\partial y} \left(k \frac{\partial T}{\partial y} \right) \right] + [W_x, W_y] \cdot \left[\frac{\partial T}{\partial x}, \frac{\partial T}{\partial y} \right]$$

Donde

$$\left\{ \begin{array}{ll} \rho & \text{Densidad de cada material} \\ \left[\frac{Kg}{m^3} \right] & \\ c_p & \text{Calor especifico de cada material} \\ \left[\frac{J}{KKg} \right] & \\ k & \text{Conductividad termica de cada material} \\ \left[\frac{W}{mK} \right] & \\ W & \text{Vector de viento} \\ \left[\frac{m}{s} \right] & \end{array} \right.$$

Ademas notar que k, ρ, c_p son variables que dependen de la posicion y de la temperatura

```
[ ]: # Config
W = np.array([0, 0])
pixel_size = 2.7
total_time = 3600 # Tiempo de la prediccion
img_at = 5        # Guarda una img para el gif final cada 5% del tiempo a
↳ simular

model = Solver(fire, Rho, Cp, K, W, pixel_size, total_time, img_at)
model.solve()
model.plot()
```

2.1 TODO

- Modelo 3d
- Forma automatica de calcular el tamaño del pixel
- Interpolar los pixel para tener resolución espacial