# DESARROLLO DE UN ALGORITMO PARA LA PRESENTACIÓN DE CAMPOS VECTORIALES PARA EL ANÁLISIS DE EXPANSIONES URBANAS

Instituto de Investigaciones y Políticas del Ambiente Construido (IIPAC CONICET-UNLP). Calle 47 n 162, CP. 1900, La Plata, Buenos Aires | Cátedra de Edificios. Departamento de Ingeniería Civil (FI UNLP).

chevezpedro@gmail.com

## 1.- Introducción

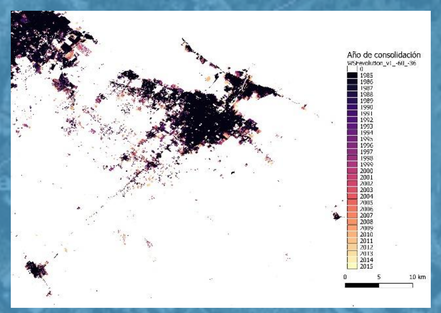
El crecimiento y la expansión urbana son analizados por diversas disciplinas tales como la geografía, la economía, el urbanismo, la ingeniería, etc.; y suele abordarse bajo el concepto de cambios de uso del suelo. Para su estudio, existen distintos tipos de modelos. Estos pueden ser estadísticos y econométricos, de interacción espacial, de optimización, integrados y otros. Dentro de esta última categoría se incluyen los modelos basados en ciencias naturales, los cuales permiten diversas aplicaciones mediante extensiones en los sistemas de información geográfica, como el análisis espacial, estadístico y de campos vectoriales.

Se identifican escasos trabajos que aborden la utilización de campos vectoriales sobre soportes SIG, en particular con un enfoque temporal. A continuación, se enumeran algunos casos, como el de Zhong et al. (2012), quienes mostraron campos vectoriales para el drenaje urbano de agua. En el trabajo de Liu et al. (2015), se propone un campo vectorial para visualizar el impulso de la demanda geográfico-temporal en los sistemas de transporte a partir de trayectorias observadas de una población. Por su parte, el trabajo de Mazzoli et al. (2019) aborda el mapeo de flujos diarios de desplazamientos al trabajo a partir de un campo vectorial. Por último, podemos mencionar el trabajo de autoría propia (Chévez y Barbero, 2017), donde se evaluó un método que utiliza técnicas manuales de recolección de datos y procesamiento para la graficación de campos vectoriales en sectores urbanos de pequeña escala.

Objetivo: el presente trabajo desarrolla un algoritmo para la representación de campos vectoriales que permitan graficar las dinámicas de expansión urbana y, así, brindar una herramienta que sirva para el análisis de este fenómeno en el territorio.

## 2.- Parte experimental: set de datos y principio de funcionamiento del algoritmo

Este análisis se basa en los datos del World Settlement Footprint (WSF) Evolution, el cual contiene mapas raster de 30 metros por píxel. Estos muestran la extensión global de los asentamientos humanos año a año desde 1985 hasta 2015. El WSF Evolution fue generado a partir de imágenes satelitales de Landsat-5 y Landsat-7, lo que permite obtener una representación detallada del crecimiento urbano a lo largo del tiempo. En la Figura 1 se observa el dataset para la ciudad de La Plata.




El propósito general del algoritmo es recorrer una imagen raster en formato .tif para analizar el patrón de expansión urbana píxel por píxel, analizando su propio año de consolidación y el de sus ocho vecinos circundantes. De esta forma, se genera un campo vectorial que representa la dirección y magnitud del crecimiento de la urbanización.

### CÓDIGO Parte A

**(A)**

```
for i in range(1, rows - 1):
    for j in range(1, cols - 1):
        central_value = img[i, j]
        if central_value > 0:
            vector_x = 0
            vector_y = 0
            valid_vectors = 0
```

**(A)** El algoritmo recorre la imagen (descarta los bordes) y, una vez identificado el primer píxel válido, se obtiene su valor y se verifica que sea mayor que 0, lo que indica que es un píxel consolidado y debe ser analizado. En caso de cumplir con esta condición, se inicializan tres variables:

- vector_x y vector_y, que almacenarán la dirección del crecimiento urbano en los ejes X e Y, respectivamente.
- valid_vectors: conteo de cuántos vecinos cumplen con los requisitos para ser considerados en el cálculo del vector de expansión.

**(B)** Se analizan los 8 píxeles vecinos al píxel central para determinar si presentan una consolidación más reciente. Se recorre la lista de 8 direcciones posibles, sumando dx y dy a las coordenadas (i, j) del píxel actual.

Antes de evaluar el vecino, se verifica que sus coordenadas (nx, ny) estén dentro de los límites de la imagen. Si el píxel vecino también es consolidado (neighbor_value > 0) y su valor es mayor que el del píxel central (neighbor_value > central_value), significa es una urbanización más reciente, es decir que la urbanización avanzó en esa dirección. En este caso, se calcula la diferencia de años entre el píxel vecino y el central (year_difference = abs(neighbor_value - central_value)).

Si esta diferencia es mayor a 0, se calcula la magnitud del vector de expansión del píxel central respecto a cada vecino usando la Ecuación 1:

$$\text{magnitud} = \frac{1}{\text{year\_difference}} * \frac{1}{(2015 - \text{neighbor\_value} + 1)} \quad (1)$$

**(C)** Se calcula la descomposición de cada vector en sus componentes X e Y, utilizando funciones trigonométricas para determinar la dirección:

- vector_x acumula la componente horizontal, usando cos(θ).
- vector_y acumula la componente vertical, usando sin(θ).

Si valid_vectors > 0, se establece el punto para posicionar el vector en el centro del píxel y se convierten las coordenadas del píxel de la imagen a coordenadas geográficas utilizando la matriz de transformación.

### CÓDIGO Parte B y C

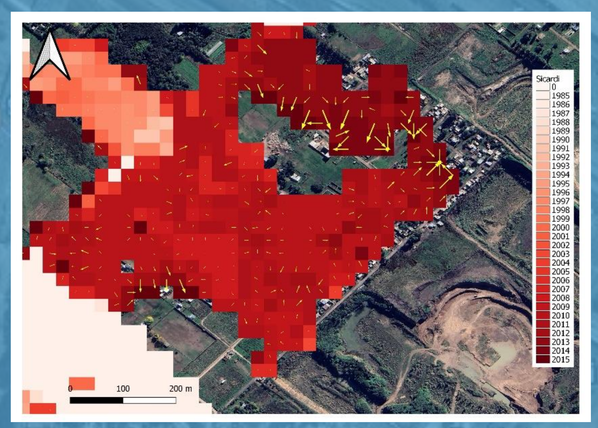
**(B)**

```
for dx, dy in [(-1, -1), (-1, 0), (-1, 1), (0, -1), (0, 1), (1, -1), (1, 0), (1, 1)]:
    nx, ny = i + dx, j + dy
    if 0 <= nx < rows and 0 <= ny < cols:
        neighbor_value = img[nx, ny]
        if neighbor_value > 0 and neighbor_value > central_value:
            year_difference = abs(neighbor_value - central_value)
            if year_difference > 0 and (2015 - neighbor_value) >= 0:
                magnitud = (1 / year_difference) * (1/(2015 - neighbor_value + 1))
```

**(C)**

```
                vector_x += magnitud * np.cos(np.arctan2(dy, dx))
                vector_y += magnitud * np.sin(np.arctan2(dy, dx))
                valid_vectors += 1
    if valid_vectors > 0:
        center_x = j + 0.5  # Centrar en el píxel
        center_y = i + 0.5  # Centrar en el píxel
        lon, lat = transform * (center_x, center_y)
        line = LineString([Point(lon, lat), Point(lon + vector_x, lat + vector_y)])
        features.append(geojson.Feature(geometry=line,
properties={"magnitude": np.sqrt(vector_x**2 + vector_y**2)}))
```

## 3.- Resultados y discusión


A partir de la evaluación del crecimiento de dos sectores urbanos, es posible visualizar que los campos vectoriales indicaron tendencias de crecimiento hacia sectores de alta vulnerabilidad ambiental, como pueden ser arroyos o canteras. En este sentido, a partir de la actualización de la información a partir de nuevos conjuntos de datos y del análisis de sectores de riesgo sería posible determinar puntos de crecimiento conflictivos para establecer medidas para reducir el impacto de posibles efectos negativos. Se espera en próximas investigaciones poder implementar el algoritmo sobre sectores urbanos más amplios y aplicarlo sobre diferentes casos de estudio a los efectos de indagar acerca de su efectividad para analizar y explicar el fenómeno social.

8º Jornadas ITEE 2025

UNIVERSIDAD NACIONAL DE LA PLATA