QR 코드 영역
QR 삽입 후
테두리 삭제

Youtube 영상 QR

해양 쓰레기 추적 및 실시간 시각화

바다를 깨끗하게 유지해보자!

3211 박연호 3302 서선민

동기

여러 뉴스 기사들을 보던 중 태평양에 형성된 남한 면적 약 16배 크기의 플라스틱 쓰레기 섬의 구성 중 일부가 일본, 중국, 한국 등에서 배출된 것으로 확인되었다는 사실을 알게 되었다. 쓰레기의 발원지를 분석한 결과 일본 34%, 중국 32%, 남북한 10%, 미국 7% 등 다양한 국가에서 쓰레기가 배출되었다는 것을 알 수 있다. 이에 전세계 해류 데이터를 실시간으로 불러와 지도에 시각화하고 각 지역의 쓰레기 이동 경로 및 그 비율을 도출하는 프로그램을 제작하여 해양쓰레기 제거에 도움이 될 수 있는 프로그램을 제작하려 한다.

융합분야 및 설계(계획)

〈정보〉
기상청과 나사 등을 통해 기상 정보가 담긴 nc 파일을 수집한 이후 파이썬의 netCDF4, matplotlib, numpy 등의 모듈을 사용하여 파일에 담긴 데이터를 분석하고 이를 활용하여 전세계 해류 데이터를 시각화한다. 이후 시각화한 데이터를 통해 각 지역의 쓰레기 이동 경로와 그 근원지를 파악할 수 있도록 한다.

〈지구과학〉
지구과학의 해류의 이동 및 각 지역에서의 크기와 세기 등의 정보를 바탕으로 하여 해류의 이동 경로를 계산하고 각 크기를 바탕으로 하여 해양 쓰레기의 예상 이동 경로 및 양을 도출할 수 있도록 한다.

프로그램의 원리

〈해류 데이터를 통해 해류 시각화〉
1. 지정한 url을 통해 원하는 데이터를 담은 .nc 형식의 파일을 불러온다.
2. 파일에 저장된 각 지역의 좌표와 그 좌표마다 지정된 해류의 속도 및 방향 데이터를 읽어온다.
3. 불러온 데이터를 통해 해류의 속도와 방향 데이터와 함께 생성할 셀의 중심 좌표를 설정한다.
4. 시간대별 데이터를 불러온 후 이를 화살표로 연결하여 방향을 표시한다.
5. 데이터를 지도에 나타내기 위해 도 단위로 바꿔준다.
6. 최종 데이터를 통해 matplotlib 모듈을 통해 지도로 출력한다.

〈해류 데이터를 바탕으로 한 해양쓰레기 근원지 파악〉
1. 조사한 해양 쓰레기섬의 좌표를 저장한다.
2. 해당 좌표와 겹친 해류를 파악한다.
3. 해류의 크기와 흐름을 통해 해류의 이동 경로를 확인하고 각 지점에서의 크기와 최종 도달 위치까지의 크기를 확인하여 이를 바탕으로 근원지 및 그 비율을 계산하여 출력한다.

작성 코드

```
import matplotlib.pyplot as plt
import numpy as np
import matplotlib.tri as Tri
import netCDF4
import datetime as dt

url = 'http://www.smast.umassd.edu:8080/thredds/dodsC/fvcom/hindcasts/30yr_gom3/mean'
# Open DAP
nc = netCDF4.Dataset(url).variables

nc['temp'].shape
nc['nv'].shape

start = dt.datetime(1991,1,1,0,0,0) + dt.timedelta(hours=+0)
start = dt.datetime(1992,7,1,0,0,0) + dt.timedelta(hours=+0)
start = dt.datetime(1992,8,1,0,0,0) + dt.timedelta(hours=+0)
start = dt.datetime(1992,6,1,0,0,0) + dt.timedelta(hours=+0)

# Get desired time step
time_var = nc['time']
itime = netCDF4.date2index(start,time_var,select='nearest')

# Get lon,lat coordinates for nodes (depth)
lat = nc['lat'][:]
lon = nc['lon'][:]
# Get lon,lat coordinates for cell centers (depth)
latc = nc['latc'][:]
lonc = nc['lonc'][:]
# Get Connectivity array
nv = nc['nv'][:].T - 1
# Get depth
h = nc['h'][:]  # depth

tm = netCDF4.num2date(time_var[itime],time_var.units)
daystr = tm.strftime('%Y-%b-%d %H:%M')

# round to nearest 10 minutes to make titles look better
tm += dt.timedelta(minutes=5)
tm -= dt.timedelta(minutes=tm.minute % 10, seconds=tm.second, microseconds=tm.microsecond)
daystr = tm.strftime('%Y-%b-%d %H:%M')

tri = Tri.Triangulation(lon,lat, triangles=nv)
```

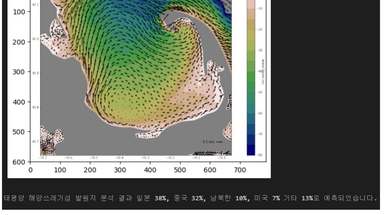
```
layer = 0
u = nc['u'][itime, ilayer, :]
v = nc['v'][itime, ilayer, :]

levels=np.arange(-30,2,1)
ax = [-70.7, -70.6, 41.48, 41.55]
ax = [-70.75, -70.6, 41.48, 41.56]
maxvel = 1.0
subsample = 2

levels=np.arange(-34,2,1)   # depth contours to plot
ax= [-70.97, -70.75, 42.15, 42.35] #
maxvel = 0.5
subsample = 3

levels=np.arange(-34,2,1)   # depth contours to plot
ax= [-70.72, -70.33, 41.25, 41.45] #
maxvel = 0.5
subsample = 1

levels=np.arange(-80,5,5)   # depth contours to plot
ax= [-70.71, -69.96, 41.66, 42.15] #
maxvel = 0.1
subsample = 1

ind = np.argwhere((lonc >= ax[0]) & (lonc <= ax[1]) & (latc >= ax[2]) & (latc <= ax[3]))

np.random.shuffle(ind)
Nvec = int(len(ind) / subsample)
idx = ind[:Nvec]

fig=plt.figure(figsize=(10,10))
plt.subplot(111,aspect=(1.0/np.cos(lat.mean()*np.pi/180.0)))
plt.tricontourf(tri, -h,levels=levels,shading='faceted',cmap=plt.cm.gist_earth)
plt.axis(ax)
plt.gca().patch.set_facecolor('0.5')
cbar = plt.colorbar()
cbar.set_label('Water Depth (m)', rotation=-90)
Q = plt.quiver(lonc[idx],latc[idx],u[idx],v[idx],scale=5)
maxstr='%3.1f m/s' % maxvel
qk = plt.quiverkey(Q,0.92,0.08,maxvel,maxstr,labelpos='N')
plt.title('NECOFS Velocity, Layer %d, %s UTC' %(ilayer, daystr))
```

파이썬의 Matplotlib, Numpy, netCDF4, datetime 등의 모듈을 사용하여 프로그램 코드를 작성하였다. .nc 확장자 형식의 파일을 활용하기 위한 netCDF4 모듈을 통해 nc 파일들을 불러오도록 하였으며 행렬이나 일반적인 대규모 다차원 배열을 처리할 때 사용하는 Numpy모듈과 Numpy 모듈을 활용한 플로팅 라이브러리인 Matplotlib 모듈을 통해 nc 파일들을 통해 불러온 데이터를 가공하고 가공을 통해 도출된 결과물을 pyplot을 통해 출력하도록 제작하였다.

실행 화면

```
img_result = img.imread("C:\\Users\\lover0\\Desktop\\asdfasf.png")
plt.imshow(img_result)
plt.show()
print(result)
```

✓ 0.3s



아쉬운 점

전세계 해류 데이터를 바탕을 하여 이를 시각화하고 쓰레기의 근원지를 예측하여 각 지역별 %로 출력하는 것이 목표였으나 개발 중 문제로 인해 출력이 잘 되지 않거나 잘못된 결과를 출력하는 경우가 많이 발생하게 되었다.

느낀 점

처음 의도했던 것과 달리 개발 과정이 순탄치 않아서 힘든 시간이 되었던 것 같고 .nc 확장 형식의 파일을 다루는 과정에서 가장 많은 문제를 맞닥뜨렸던 것 같다. 프로그램을 안정화하고 완성하지 못한 것 같아 아쉬웠지만 이후 이를 확장시킨 프로그램을 제작하여 현재의 문제점들을 해결하고 싶다.