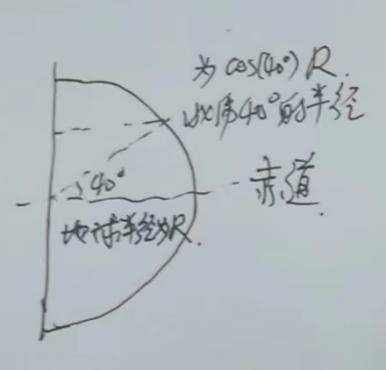
**Dead\_reckoning Python Code**

import os.path as osp  
import numpy as np  
import matplotlib.pyplot as plt  
from imageio import imread  
  
def load\_data(lat\_lng\_fname, yaw\_fname, groundtruth\_fname):  
 lat, lng = np.loadtxt(lat\_lng\_fname, delimiter=',').T  
 yaw = np.loadtxt(yaw\_fname, delimiter=',')  
 groundtruth = np.loadtxt(groundtruth\_fname, delimiter=',')  
 return lat, lng, yaw, groundtruth  
def gen\_xy\_from\_lat\_lng(lat, lng, circ):  
 lat\_adj = lat - lat[0] # 纬度，纬度沿着经度的线进行变化  
 lng\_adj = lng - lng[0] # 经度，经度沿着纬度的线进行变化

x = -lng\_adj \* (circ \* np.cos(lat) / 360) # 原理见下图；  
 y = +lat\_adj \* (circ / 360)  
 return x, y

  
def rotate(x, y, theta):  
 c, s = np.cos(theta), np.sin(theta)  
 rotation = [[c, -s], [s, c]]  
 x, y = rotation @ np.c\_[x, y].T

# x和y都是n\*1维的向量，二者组成n组坐标；np.c\_用于连接两个同维度的矩阵为一个矩阵；

# @为python的乘法运算单元；  
 return x, y  
def gen\_xy\_from\_yaw(yaw, dt, v0):  
 psi0 = -np.pi / 2 + yaw[0] # 用弧度表示的偏航角  
 psiR = yaw  
 dxdt = v0 \* np.sin(psi0 - psiR) # psi0 - psiR表示弧度变化量，本公式求得x方向的速度  
 dydt = v0 \* np.cos(psi0 - psiR) # psi0 - psiR表示弧度变化量，本公式求得y方向的速度  
 x = (dt \* dxdt).cumsum() # python中的函数，对以前的数据进行累加  
 y = (dt \* dydt).cumsum()  
 return x, y  
def shift\_and\_scale(x0, y0, x1, y1):  
 ux = x1.mean() - x0.mean()  
 uy = y1.mean() - y0.mean()  
 alpha\_x = (x0.max() - x0.min()) / (x1.max() - x1.min())  
 alpha\_y = (y0.max() - y0.min()) / (y1.max() - y1.min())  
 x1 = alpha\_x \* (x1 - ux)  
 y1 = alpha\_y \* (y1 - uy)  
 return x1, y1  
def configure\_axis(axis, title):  
 axis.set\_xticklabels([])  
 axis.set\_yticklabels([])  
 axis.tick\_params(axis='both', which='both', length=0)  
 axis.set\_xlabel('East')  
 axis.set\_ylabel('North')  
 axis.set\_title(title)  
def plot\_navigation(x0, y0, x1, y1, groundtruth, img):  
 x2, y2 = groundtruth.T  
 fig, axarr = plt.subplots(nrows=1, ncols=3, figsize=(12, 4))  
 titles = ['GPS', 'Magnetometer', 'Google Maps']  
 axarr[0].plot(x0, y0, c='black')  
 axarr[1].plot(x1, y1, c='black')  
 axarr[2].plot(x2, y2, c='orange', linestyle='dashed')  
 axarr[2].imshow(img, aspect='auto')  
 for i in range(3):  
 configure\_axis(axarr[i], titles[i])  
 fig.tight\_layout()  
 plt.savefig('./data/dead-reckoningttt.png')  
def main():  
 circ = 40.075 \* 1e6 # Circumference of the Earth (m) 地球周长4万多公里  
 vel = 0.97 # Measured walking speed (m/s) 步行速度 0.97m/s  
 dt = 0.02 # Interval between yaw samples (s) 磁力计航向角的采样周期20ms  
 fpaths = ['./data/lat-lng.csv', './data/yaw.csv', './data/groundtruth.csv']  
 lat, lng, yaw, groundtruth = load\_data(\*fpaths)

# latitude表示纬度，longitude表示经度，yaw是偏航角；这三个量的单位都是角度；  
 x0, y0 = gen\_xy\_from\_lat\_lng(lat, lng, circ)  
 x0, y0 = rotate(x0, y0, -np.pi / 9)

# x0，y0为从GPS获得的经纬度数据计算获得的轨迹  
 x1, y1 = gen\_xy\_from\_yaw(yaw, dt, vel)  
 x1, y1 = shift\_and\_scale(x0, y0, x1, y1)

# x1，y1为从磁力计获得的航向角数据计算获得的轨迹  
 img = imread('./data/upenn.png')  
 plot\_navigation(x0, y0, x1, y1, groundtruth, img)  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 main()