Proof of concept: Calculate sun angle

Shadow length and time of day have been collected from http://centrodedescargas.cnig.es. Check whether these can calculate a hub height that approximately matches the identified hub heights.

Conclusions:

1. The aerial photos have a timestamp in UTC
2. The quality of initial spanish metadata is awful (3 out of 7 are a match)
3. 5m is a reasonable accuracy goal (weak evidence, based on three matches)

In [66]:
```
import os
from datetime import datetime

import dotenv
import numpy as np
import pandas as pd
from skyfield import almanac, api

dotenv.load_dotenv('../.env')
dotenv.load_dotenv('../.env.secret')

sites = pd.read_csv('../data/poc_measurements.csv')
```

In [67]:
```
sites = sites.assign(
    date_string = lambda x: x.date + ' ' + x.hora + os.environ.get('utc_offset'),
    datetime_utc = lambda x: pd.to_datetime(x.date_string, dayfirst=True, utc=True)
)
sites
```

Out[67]:

	site	latitude	longitude	num_turbines	hub_height	orthophoto_file	photo_file	pn
0	adrano	42.914466	-9.038402	36	50.0	NaN	h50_0093_fot_052-02924	PNOA_202
1	adrano	42.914466	-9.038402	36	50.0	NaN	h50_0093_fot_053-02870	PNOA_202
2	brulles	42.542000	-3.926000	20	60.0	NaN	h50_0166_fot_018-0049	PNOA_20
3	brulles	42.542000	-3.926000	20	60.0	NaN	h50_0166_fot_017-0050	PNOA_20
4	brulles	42.542000	-3.926000	20	60.0	PNOA_MA_OF_ETRS89_HU30_h50_0166	NaN	
5	cabimonteros_escurrillo	42.247566	-2.254341	75	45.0	NaN	h50_0242_fot_047-82915	PNOA_2
6	caluengo_esteban	42.319624	-1.832122	33	61.0	NaN	h50_0244_fot_099-88226	PNOA_2
7	cerros_radona	41.268411	-2.424675	16	100.0	NaN	h50_0435_fot_163-0025	PNOA_20
8	cortijo_guerra	36.530000	-6.190000	17	80.0	NaN	NaN	
9	cortijo_guerra	36.499251	-6.074016	14	105.0	NaN	h50_1069_fot_041-5666	PNOA_2019
10	fatarella	41.175145	0.499572	21	80.0	NaN	h50_0444_fot_156-0026_cog	PNOA_2021_
11	gatun_larriba	42.151374	-2.041051	11	61.5	NaN	h50_0281_fot_105-98119	PNOA_2
12	gatun_larriba	42.151374	-2.041051	33	61.5	NaN	NaN	
13	igea_cornago	42.014288	-2.067785	14	67.0	NaN	NaN	
14	lorenzo	41.770000	-5.040000	47	105.0	NaN	NaN	
15	lorenzo	41.679642	-5.003983	33	105.0	NaN	NaN	
16	lorenzo_bajoz	41.770000	-5.040000	47	105.0	NaN	NaN	
17	lorenzo_bajoz	41.679642	-5.003983	33	105.0	NaN	NaN	
18	ourol	43.560394	-7.660003	22	109.0	NaN	h50_0008_fot_013-86552	PNOA_202
19	palomarejo	37.459574	-5.149760	15	78.0	NaN	h50_0987_fot_075-6928	PNOA_2019
20	paramo_poza	42.670000	-3.570000	133	55.0	NaN	NaN	
21	paxareiras_montevos	42.840000	-9.075000	66	35.0	NaN	NaN	
22	paxareiras_y	42.840000	-9.075000	66	35.0	NaN	NaN	
23	puntal	37.169838	-4.846448	10	67.0	NaN	NaN	
24	santo_cristo	41.795597	-1.366295	20	80.0	NaN	NaN	
25	sasdonigas	43.431497	-7.372017	5	87.0	NaN	NaN	
26	tella	38.479390	-1.324162	25	90.0	NaN	NaN	
27	toranza	41.718260	-1.917157	30	45.0	NaN	NaN	
28	torre_madrina	41.097632	0.228387	20	100.0	NaN	NaN	
29	valcaire	36.998491	-3.681562	8	44.0	NaN	NaN	
30	valdeconejos	40.777637	-0.862297	38	55.0	NaN	NaN	
31	valdivia	37.140233	-5.143969	19	80.0	NaN	NaN	
32	valiente	42.028651	-0.672390	10	93.0	NaN	NaN	
33	viudo	39.929564	-0.926380	13	78.0	NaN	NaN	

In [68]:
```
ephemeris = api.load('de421.bsp')
earth, sun = ephemeris['earth'], ephemeris['sun']
test_sites = sites[sites.hora.isna()].drop_duplicates(subset='site', keep='last')
for i, site in test_sites.iterrows():
    observer = earth + api.wgs84.latlon(latitude_degrees=site.latitude, longitude_degrees=site.longitude)
    time = api.load.timescale().from_datetime(site.datetime_utc)
    altitude, _, _ = observer.at(time).observe(sun).apparent().altaz()
    test_sites.loc[i, ['altitude_degrees', 'altitude_radians']]= altitude.degrees, altitude.radians
test_sites = test_sites.assign(
    estimated_hub_height=lambda x: np.tan(x.altitude_radians) * x.shadow_length
)
test_sites[[
    'site', 'latitude', 'longitude', 'hub_height', 'shadow_length', 'altitude_degrees', 'estimated_hub_height'
]].round(2)
```

Out[68]:

	site	latitude	longitude	hub_height	shadow_length	altitude_degrees	estimated_hub_height
1	adrano	42.91	-9.04	50.0	70.4	65.26	152.78
4	brulles	42.54	-3.93	60.0	86.8	31.28	52.73
6	caluengo_esteban	42.32	-1.83	61.0	75.7	22.36	31.13
9	cortijo_guerra	36.50	-6.07	105.0	29.7	53.59	40.26
10	fatarella	41.18	0.50	80.0	52.4	51.95	66.95
18	ourol	43.56	-7.66	109.0	46.0	36.18	33.64
19	palomarejo	37.46	-5.15	78.0	23.6	62.73	45.79

In [69]:
```
# When is the solar noon in Adrano?
turbine = api.wgs84.latlon(site.latitude, site.longitude)
t1 = api.load.timescale().utc(2019, 7, 14)
t2 = api.load.timescale().utc(2019, 7, 15)
f = almanac.meridian_transits(ephemeris, sun, turbine)
times, events = almanac.find_discrete(t1, t2, f)
times[1].tt_calendar()
```

Out[69]:
```
(2019, 7, 14, 12, 27, 37.551741600036624)
```

In [70]:
```
# How does the altitude angle for a day in June in Adrano?
angle_list = []
observer = earth + api.wgs84.latlon(latitude_degrees=site.latitude, longitude_degrees=site.longitude)
site = sites.iloc[0]
for hour in range(24):
    time = api.load.timescale().from_datetime(site.datetime_utc.replace(hour=hour))
    altitude, _, _ = observer.at(time).observe(sun).apparent().altaz()
    angle_list.append({'hour': hour, 'degrees': altitude.degrees, 'radians': altitude.radians})
pd.DataFrame(angle_list)
```

Out[70]:

	hour	degrees	radians
0	0	-28.912091	-0.504611
1	1	-28.419238	-0.496009
2	2	-24.801472	-0.432867
3	3	-18.540774	-0.323598
4	4	-10.273396	-0.179305
5	5	-0.577835	-0.010085
6	6	10.098414	0.176251
7	7	21.425782	0.373950
8	8	33.150644	0.578588
9	9	45.040763	0.786110
10	10	56.787406	0.991127
11	11	67.691367	1.181437
12	12	75.263078	1.313589
13	13	73.666129	1.285716
14	14	64.590747	1.127321
15	15	53.304730	0.930343
16	16	41.469972	0.723788
17	17	29.602802	0.516666
18	18	17.972368	0.313677
19	19	6.810266	0.118862
20	20	-3.612174	-0.063044
21	21	-12.933795	-0.225737
22	22	-20.666643	-0.360701
23	23	-26.201134	-0.457296

In [71]:
```
# Which utc_offset minimises the errors in the estimated hub height?
result_list = []
for _, site in test_sites.iterrows():
    observer = earth + api.wgs84.latlon(latitude_degrees=site.latitude, longitude_degrees=site.longitude)
    site_result_dict = {}
    for utc_offset in range(3):
        date_string = datetime.strptime(
            f'{site.date} {site.hora} +{utc_offset:02}00', '%d/%m/%Y %H:%M:%S %z')
        time = api.load.timescale().from_datetime(date_string)
        altitude, _, _ = observer.at(time).observe(sun).apparent().altaz()
        estimated_hub_height = round(np.tan(altitude.radians) * site.shadow_length, 1)
        site_result_dict[f'estimate_{utc_offset}'] = estimated_hub_height

    result_list.append(site_result_dict)

results = pd.concat([
    test_sites[['site', 'hub_height', 'shadow_length']],
    pd.DataFrame(result_list, index = test_sites.index)]
    , axis=1
)
results['error_0'] = results.estimate_0 - results.hub_height
results
```

Out[71]:

	site	hub_height	shadow_length	estimate_0	estimate_1	estimate_2	error_0
1	adrano	50.0	70.4	72.3	105.4	152.8	22.3
4	brulles	60.0	86.8	107.1	77.4	52.7	47.1
6	caluengo_esteban	61.0	75.7	61.3	46.6	31.1	0.3
9	cortijo_guerra	105.0	29.7	104.6	63.7	40.3	-0.4
10	fatarella	80.0	52.4	143.9	99.6	66.9	63.9
18	ourol	109.0	46.0	69.9	49.0	33.6	-39.1
19	palomarejo	78.0	23.6	81.2	71.2	45.8	3.2

In [72]:
```
summary_list = []
for utc_offset in range(3):
    errors = results[f'estimate_{utc_offset}'] - results.hub_height
    summary_list.append({
        'utc_offset': utc_offset,
        'within_5m': errors.between(-5, 5).sum(),
        'within_10m': errors.between(-10, 10).sum(),
        'within_20m': errors.between(-20, 20).sum()
    })
pd.DataFrame(summary_list)
```

Out[72]:

	utc_offset	within_5m	within_10m	within_20m
0	0	3	3	3
1	1	0	1	4
2	2	0	1	2

In [73]:
```

```